Red Hat Enterprise Linux 6 Verwaltung des Load Balancer Add-Ons

---

Load Balancer Add-on für Red Hat Enterprise Linux
Ausgabe 6

Red Hat Enterprise Linux 6 Verwaltung des Load Balancer Add-Ons

Load Balancer Add-on für Red Hat Enterprise Linux
Ausgabe 6

Rechtlicher Hinweis

Copyright © 2014 Red Hat, Inc.

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Zusammenfassung

Ein System mit dem Load Balancer Add-On bietet eine hochverfügbare und skalierbare Lösung für Produktionsservices, unter Verwendung spezieller Linux Virtual Server (LVS) für das Routing und für Lastverteilungstechniken. Dieses Handbuch behandelt die Konfiguration von Hochleistungssystemen und -diensten mit Red Hat Enterprise Linux und dem Load Balancer Add-On für Red Hat Enterprise Linux 6.

Inhaltsverzeichnis

Einführung **3**
1. Dokumentkonventionen 3
1.1. Typografische Konventionen 4
1.2. Konventionen für Seitenansprachen 5
1.3. Anmerkungen und Warnungen 6
2. Feedback 6

Kapitel 1. Überblick über das Load Balancer Add-On **8**
1.1. Eine grundlegende Load Balancer Add-On Konfiguration 8
1.1.1. Daten replizieren und gemeinsam verwenden auf realen Servern 10
1.1.1.1. Konfigurieren von realen Servern zur Synchronisierung von Daten 10
1.2. Eine dreischichtige Load Balancer Add-On Konfiguration 10
1.3. Übersicht über das Load Balancer Add-On Scheduling 11
1.3.1. Scheduling-Algorithmen 12
1.3.2. Server-Gewichtung und Scheduling 13
1.4. Routing-Methoden 14
1.4.1. NAT-Routing 14
1.4.2. Direktes Routing 16
1.4.2.1. Direktes Routing und die ARP-Einschränkung 16
1.5. Persistenz und Firewall-Markierungen 17
1.5.1. Persistenz 17
1.5.2. Firewall-Markierungen 18
1.6. Load Balancer Add-On – Ein Blockdiagramm 18
1.6.1. Load Balancer Add-On Komponenten 19
1.6.1.1. pulse 20
1.6.1.2. lvs 20
1.6.1.3. ipvsadm 20
1.6.1.4. nanny 20
1.6.1.5. /etc/sysconfig/ha/lvs.cf 20
1.6.1.6. Piranha-Konfigurationstool 20
1.6.1.7. send_arp 20

Kapitel 2. Erste Konfigurationsschritte für das Load Balancer Add-On **22**
2.1. Konfigurieren von Diensten auf dem LVS-Router 22
2.2. Einrichten eines Passworts für das Piranha-Konfigurationstool 23
2.3. Starten des Piranha-Konfigurationstool-Dienstes 23
2.3.1. Konfigurieren des Piranha-Konfigurationstool-Webserver-Ports 24
2.4. Einschränken des Zugriffs auf das Piranha-Konfigurationstool 24
2.5. Aktivieren der Paketweiterleitung 25
2.6. Konfigurieren von Diensten auf den realen Servern 25

Kapitel 3. Einrichten des Load Balancer Add-Ons **27**
3.1. Das NAT Load Balancer Add-On Netzwerk 27
3.1.1. Konfigurieren von Netzwerkschnittstellen für Load Balancer Add-On mit NAT 27
3.1.2. Routing auf den realen Servern 29
3.1.3. Aktivieren von NAT-Routing auf den LVS-Routern 29
3.2. Load Balancer Add-On mit direktem Routing 30
3.2.1. Direktes Routing und arptables_jf 31
3.2.2. Direktes Routing und iptables 32
3.3. Zusammenstellen der Konfiguration 33
3.3.1. Allgemeine Tipps für ein Netzwerk mit Load Balancer Add-On 33
3.3.1.1. Suche und Bereinigung von Fehlern mit virtuellen IP-Adressen 34

3.4. Multi-Port-Dienste und Load Balancer Add-On 34
3.4.1. Zuweisen von Firewall-Markierungen 35
3.5. Konfigurieren von FTP 36
3.5.1. Funktionsweise von FTP 36
3.5.2. Auswirkungen auf das Load Balancer Add-On Routing 37
3.5.3. Erstellen von Netzwerkpaketfilterregeln 37
3.5.3.1. Regeln für aktive Verbindungen 37
3.5.3.2. Regeln für passive Verbindungen 38
3.6. Speichern der Netzwerkpaketfiltereinstellungen 39

Kapitel 4. Konfigurieren des Load Balancer Add-Ons mit dem Piranha-Konfigurationstool .. 40
4.1. Notwendige Software 40
4.2. Anmelden beim Piranha-Konfigurationstool 40
4.3. CONTROL/MONITORING 41
4.4. GLOBAL SETTINGS 43
4.5. REDUNDANCY 45
4.6. VIRTUAL SERVERS 48
4.6.1. Der Unterabschnitt VIRTUAL SERVER 49
4.6.2. Der Unterabschnitt REAL SERVER 52
4.6.3. Der Unterabschnitt EDIT MONITORING SCRIPTS 55
4.7. Synchronisieren von Konfigurationsdateien 57
4.7.1. Synchronisieren von lvs.cf 57
4.7.2. Synchronisieren von sysctl 58
4.7.3. Synchronisieren von Regeln zur Netzwerkpaketfilterung 58
4.8. Starten des Load Balancer Add-Ons 59

Verwenden des Load Balancer Add-Ons mit dem High Availability Add-On 60

Versionsgeschichte 62

Stichwortverzeichnis 62

Einführung

Dieses Handbuch liefert Informationen über die Installation, Konfiguration und Verwaltung der Load Balancer Add-On Komponenten. Das Load Balancer Add-On ermöglicht die Lastverteilung mittels spezieller Routing-Techniken, die den Datenverkehr an eine Gruppe von Servern leiten.

Die Zielgruppe dieses Handbuchs sollte bereits über umfassende Kenntnisse von Red Hat Enterprise Linux verfügen und die Konzepte von Clustern, Storage und Serverrechnern verstehen.

Dieses Dokument ist folgendermaßen aufgebaut:

- Kapitel 1, *Überblick über das Load Balancer Add-On*
- Kapitel 2, *Erste Konfigurationsschritte für das Load Balancer Add-On*
- Kapitel 3, *Einrichten des Load Balancer Add-Ons*
- Kapitel 4, *Konfigurieren des Load Balancer Add-Ons mit dem* ***Piranha-Konfigurationstool***
- Anhang A, *Verwenden des Load Balancer Add-Ons mit dem High Availability Add-On*

Weitere Informationen über Red Hat Enterprise Linux 6 finden Sie in den folgenden Quellen:

- *Red Hat Enterprise Linux Installationshandbuch* – Liefert Informationen bezüglich der Installation von Red Hat Enterprise Linux 6.
- *Red Hat Enterprise Linux Bereitstellungshandbuch* – Liefert Informationen bezüglich der Implementierung, der Konfiguration und der Administration von Red Hat Enterprise Linux 6.

Weitere Informationen über das Load Balancer Add-On und zugehörige Produkte für Red Hat Enterprise Linux 6 finden Sie in den folgenden Quellen:

- *Überblick über die Red Hat Cluster Suite* – Liefert einen allgemeinen Überblick über das High Availability Add-On, das Resilient Storage Add-On und das Load Balancer Add-On.
- *Konfiguration und Verwaltung des High Availability Add-Ons* Liefert Informationen zur Konfiguration und Verwaltung des High Availability Add-Ons (auch Red Hat Cluster genannt) für Red Hat Enterprise Linux 6.
- *Administration des Logical Volume Managers* – Liefert eine Beschreibung des Logical Volume Managers (LVM), inklusive Informationen zum Einsatz von LVM in einer Cluster-Umgebung.
- *Global File System: Konfiguration und Administration* – Liefert Informationen zur Installation, Konfiguration und Wartung des Red Hat Resilient Storage Add-Ons (auch Red Hat Global File System 2 genannt).
- *DM-Multipath* – Liefert Informationen zur Verwendung des Device-Mapper-Multipath-Features von Red Hat Enterprise Linux 6.
- *Versionshinweise* – Liefert Informationen zu aktuellen Releases von Red Hat Produkten.

Dieses Dokument und andere Red Hat Dokumente stehen als HTML-, PDF-, und EPUB-Versionen online unter http://access.redhat.com/documentation/docs zur Verfügung.

1. Dokumentkonventionen

Dieses Handbuch verwendet mehrere Konventionen, um bestimmte Wörter und Sätze hervorzuheben und Aufmerksamkeit auf bestimmte Informationen zu lenken.

1.1. Typografische Konventionen

Es werden vier typografische Konventionen verwendet, um die Aufmerksamkeit auf bestimmte Wörter und Sätze zu lenken. Diese Konventionen und die Umstände, unter denen sie auftreten, sind folgende:

`Nichtproportional Fett`

Dies wird verwendet, um Systemeingaben hervorzuheben, einschließlich Shell-Befehle, Dateinamen und -pfade. Es wird ebenfalls zum Hervorheben von Tasten und Tastenkombinationen verwendet. Zum Beispiel:

> Um den Inhalt der Datei **`my_next_bestselling_novel`** in Ihrem aktuellen Arbeitsverzeichnis zu sehen, geben Sie den Befehl **`cat my_next_bestselling_novel`** in den Shell-Prompt ein und drücken Sie **`Enter`**, um den Befehl auszuführen.

Das oben aufgeführte Beispiel beinhaltet einen Dateinamen, einen Shell-Befehl und eine Taste. Alle werden nichtproportional fett dargestellt und alle können, dank des Kontextes, leicht unterschieden werden.

Tastenkombinationen unterscheiden sich von einzelnen Tasten durch das Pluszeichen, das die einzelnen Teile einer Tastenkombination miteinander verbindet. Zum Beispiel:

> Drücken Sie **`Enter`**, um den Befehl auszuführen.
>
> Drücken Sie **`Strg`**+**`Alt`**+**`F2`**, um zu einem virtuellen Terminal zu wechseln.

Das erste Beispiel hebt die zu drückende Taste hervor. Das zweite Beispiel hebt eine Tastenkombination hervor: eine Gruppe von drei Tasten, die gleichzeitig gedrückt werden müssen.

Falls Quellcode diskutiert wird, werden Klassennamen, Methoden, Funktionen, Variablennamen und Rückgabewerte, die innerhalb eines Abschnitts erwähnt werden, wie oben gezeigt **`nichtproportional fett`** dargestellt. Zum Beispiel:

> Zu dateiverwandten Klassen zählen **`filesystem`** für Dateisysteme, **`file`** für Dateien und **`dir`** für Verzeichnisse. Jede Klasse hat ihren eigenen Satz an Berechtigungen.

Proportional Fett

Dies kennzeichnet Wörter oder Sätze, die auf einem System vorkommen, einschließlich Applikationsnamen, Text in Dialogfeldern, beschriftete Schaltflächen, Bezeichnungen für Auswahlkästchen und Radio-Buttons, Überschriften von Menüs und Untermenüs. Zum Beispiel:

> Wählen Sie **System** → **Einstellungen** → **Maus** in der Hauptmenüleiste aus, um die **Mauseinstellungen** zu öffnen. Wählen Sie im Reiter **Tasten** auf das Auswahlkästchen **Mit links bediente Maus** und anschließend auf **Schließen**, um die primäre Maustaste von der linken auf die rechte Seite zu ändern (d.h., um die Maus auf Linkshänder anzupassen).
>
> Um ein Sonderzeichen in eine **gedit**-Datei einzufügen, wählen Sie **Anwendungen** → **Zubehör** → **Zeichentabelle** aus der Hauptmenüleiste. Wählen Sie als Nächstes **Suchen** → **Suchen** aus der Menüleiste der **Zeichentabelle**, geben Sie im Feld **Suchbegriff** den Namen des Zeichens ein und klicken Sie auf **Weitersuchen**. Das gesuchte Zeichen wird daraufhin in der **Zeichentabelle** hervorgehoben. Doppelklicken Sie auf dieses hervorgehobene Zeichen, um es in das Feld **Zu kopierender Text** zu übernehmen und klicken Sie anschließend auf die Schaltfläche **Kopieren**. Gehen Sie nun zurück in Ihr Dokument und wählen Sie **Bearbeiten** → **Einfügen** aus der **gedit**-Menüleiste.

Der oben aufgeführte Text enthält Applikationsnamen, systemweite Menünamen und -elemente, applikationsspezifische Menünamen sowie Schaltflächen und Text innerhalb einer grafischen Oberfläche. Alle werden proportional fett dargestellt und sind anhand des Kontextes unterscheidbar.

Nichtproportional Fett Kursiv oder ***Proportional Fett Kursiv***

Sowohl bei nichtproportional fett als auch bei proportional fett weist ein zusätzlicher Kursivdruck auf einen ersetzbaren oder variablen Text hin. Kursivdruck kennzeichnet Text, der nicht wörtlich eingeben wird, oder angezeigten Text, der sich abhängig von den gegebenen Umständen unterscheiden kann. Zum Beispiel:

> Um sich mit einer Remote-Maschine via SSH zu verbinden, geben Sie an einem Shell-Prompt **ssh *username@domain.name*** ein. Falls die Remote-Maschine **example.com** ist und Ihr Benutzername auf dieser Maschine John lautet, geben Sie also **ssh john@example.com** ein.
>
> Der Befehl **mount -o remount *file-system*** hängt das angegebene Dateisystem wieder ein. Um beispielsweise das **/home**-Dateisystem wieder einzuhängen, verwenden Sie den Befehl **mount -o remount /home**.
>
> Um die Version des derzeit installierten Pakets zu sehen, verwenden Sie den Befehl **rpm -q *package***. Die Ausgabe sieht wie folgt aus: ***package-version-release***.

Beachten Sie die kursiv dargestellten Begriffe oben — username, domain.name, file-system, package, version und release. Jedes Wort ist ein Platzhalter entweder für Text, den Sie für einen Befehl eingeben, oder für Text, der vom System angezeigt wird.

Neben der Standardbenutzung für die Darstellung des Titels eines Werks zeigt der Kursivdruck auch die erstmalige Verwendung eines neuen und wichtigen Begriffs an. Zum Beispiel:

> Publican ist ein *DocBook* Publishing-System.

1.2. Konventionen für Seitenansprachen

Ausgaben des Terminals und Auszüge aus dem Quellcode werden visuell vom umliegenden Text hervorgehoben durch sogenannte Seitenansprachen (auch Pull-Quotes genannt).

Eine an das Terminal gesendete Ausgabe wird in den Schrifttyp **nichtproportional Roman** gesetzt und wie folgt dargestellt:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Auszüge aus dem Quellcode werden ebenfalls in den Schrifttyp **nichtproportional Roman** gesetzt, doch wird zusätztlich noch die Syntax hervorgehoben:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                  struct kvm_assigned_pci_dev *assigned_dev)
{
         int r = 0;
         struct kvm_assigned_dev_kernel *match;

         mutex_lock(&kvm->lock);

         match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                       assigned_dev->assigned_dev_id);
         if (!match) {
```

```
                printk(KERN_INFO "%s: device hasn't been assigned
before, "
                  "so cannot be deassigned\n", __func__);
                r = -EINVAL;
                goto out;
        }

        kvm_deassign_device(kvm, match);

        kvm_free_assigned_device(kvm, match);

out:
        mutex_unlock(&kvm->lock);
        return r;
}
```

1.3. Anmerkungen und Warnungen

Zu guter Letzt verwenden wir drei visuelle Stile, um die Aufmerksamkeit auf Informationen zu lenken, die andernfalls vielleicht übersehen werden könnten.

Anmerkung

Eine Anmerkung ist ein Tipp, ein abgekürztes Verfahren oder ein alternativer Ansatz für die vorliegende Aufgabe. Das Ignorieren von Anmerkungen sollte keine negativen Auswirkungen haben, aber Sie verpassen so vielleicht einen Trick, der Ihnen das Leben vereinfachen könnte.

Wichtig

Die Wichtig-Schaukästen lenken die Aufmerksamkeit auf Dinge, die sonst leicht übersehen werden können: Konfigurationsänderungen, die nur für die aktuelle Sitzung gelten oder Dienste, für die ein Neustart nötig ist, bevor eine Aktualisierung wirksam wird. Das Ignorieren von Wichtig-Schaukästen würde keinen Datenverlust verursachen, kann aber unter Umständen zu Ärgernissen und Frustration führen.

Warnung

Eine Warnung sollte nicht ignoriert werden. Das Ignorieren von Warnungen führt mit hoher Wahrscheinlichkeit zu Datenverlust.

2. Feedback

Falls Sie einen Fehler in diesem Handbuch finden oder eine Idee haben, wie dieses verbessert werden könnte, freuen wir uns über Ihr Feedback! Bitte reichen Sie einen Fehlerbericht in Bugzilla (http://bugzilla.redhat.com/bugzilla/) für das Produkt **Red Hat Enterprise Linux 6**, die Komponente **doc-Load_Balancer_Administration** und die Versionsnummer 6.1 ein.

Falls Sie uns einen Vorschlag zur Verbesserung der Dokumentation senden möchten, sollten Sie

hierzu möglichst genaue Angaben machen. Wenn Sie einen Fehler gefunden haben, geben Sie bitte die Nummer des Abschnitts und einen Ausschnitt des Textes an, damit wir diesen leicht finden können.

Kapitel 1. Überblick über das Load Balancer Add-On

Anmerkung

Ab Red Hat Enterprise Linux 6.6 bietet Red Hat Unterstützung für HAProxy und keepalived zusätzlich zur Piranha-Lastverteilungssoftware. Informationen über die Konfiguration eines Red Hat Enterprise Linux Systems mit HAProxy und keepalived finden Sie in der Dokumentation zur Lastverteilungsadministration für Red Hat Enterprise Linux 7.

Das Load Balancer Add-On ist eine Gruppe integrierter Softwarekomponenten, die Linux Virtual Servers (LVS) zur Verteilung von IP-Lasten auf eine Reihe von realen Servern bereitstellen. Das Load Balancer Add-On läuft auf einem *aktiven LVS-Router* sowie auf einem *Backup-LVS-Router*. Der aktive LVS-Router hat zwei Aufgaben:

- Gleichmäßige Verteilung der Auslastung unter den realen Servern.
- Überprüfung der Integrität der Dienste auf jedem realen Server.

Der Backup-LVS-Router überwacht den aktiven LVS-Router und übernimmt dessen Aufgaben im Falle eines Ausfalls.

Dieses Kapitel liefert einen Überblick über die Komponenten und Funktionen des Load Balancer Add-Ons und besteht aus den folgenden Abschnitten:

- Abschnitt 1.1, „Eine grundlegende Load Balancer Add-On Konfiguration“
- Abschnitt 1.2, „Eine dreischichtige Load Balancer Add-On Konfiguration“
- Abschnitt 1.3, „Übersicht über das Load Balancer Add-On Scheduling“
- Abschnitt 1.4, „Routing-Methoden“
- Abschnitt 1.5, „Persistenz und Firewall-Markierungen“
- Abschnitt 1.6, „Load Balancer Add-On – Ein Blockdiagramm“

1.1. Eine grundlegende Load Balancer Add-On Konfiguration

Abbildung 1.1, „Eine grundlegende Load Balancer Add-On Konfiguration“ zeigt eine einfache Load Balancer Add-On Konfiguration bestehend aus zwei Schichten. Auf der ersten Schicht befindet sich ein aktiver und ein Backup-LVS-Router. Jeder LVS-Router hat zwei Netzwerkschnittstellen, eine zum Internet und eine zum privaten Netzwerk, sodass sie den Datenverkehr zwischen den zwei Netzwerken steuern können. In diesem Beispiel verwendet der aktive Router *Network Address Translation* oder kurz *NAT*, um Datenverkehr vom Internet an eine variable Anzahl von realen Servern auf der zweiten Schicht zu leiten, die die notwendigen Dienste bereitstellen. Somit sind die realen Server in diesem Beispiel alle mit einem dedizierten privaten Netzwerksegment verbunden und leiten sämtlichen öffentlichen Datenverkehr durch den aktiven LVS-Router. Nach außen hin erscheinen die Server als eine einzige Einheit.

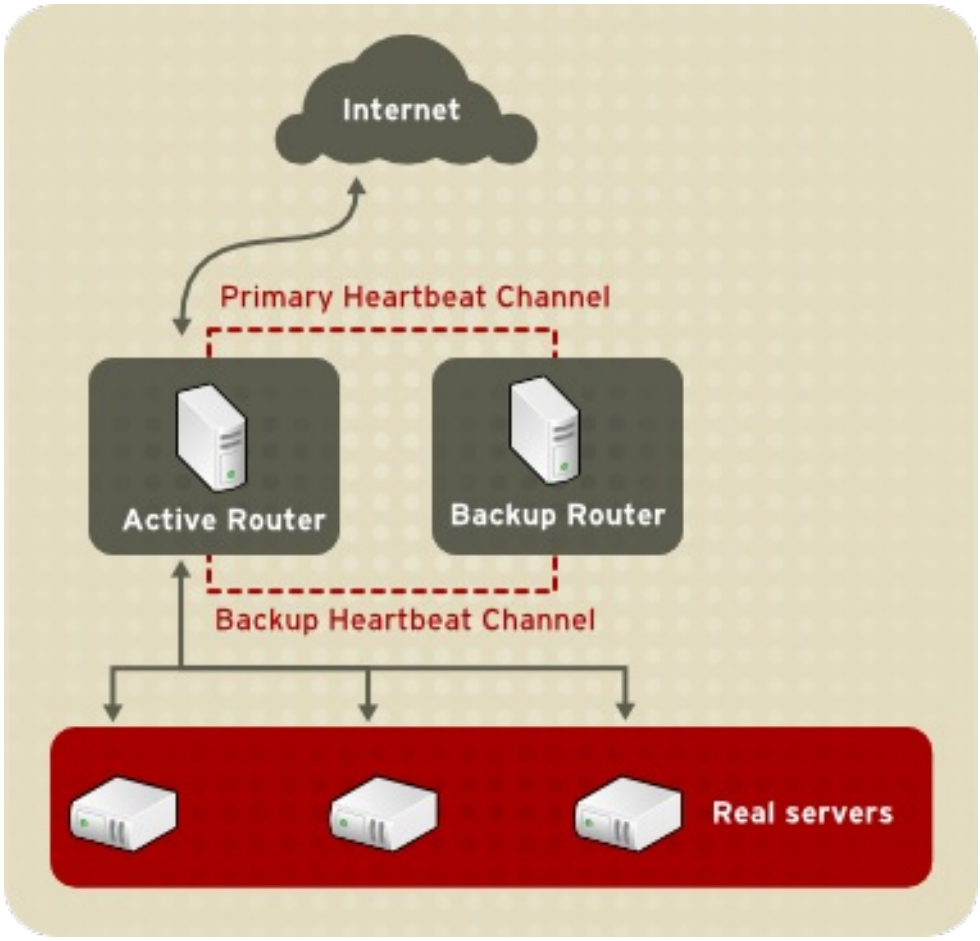


Abbildung 1.1. Eine grundlegende Load Balancer Add-On Konfiguration

Dienstanfragen, die den LVS-Router erreichen, sind an eine *virtuelle IP-Adresse* oder kurz *VIP* adressiert. Dies ist eine öffentlich routbare Adresse, die der Administrator des Netzwerks mit einem vollqualifizierten Domainnamen wie z. B. www.example.com verknüpft und die mit einem oder mehreren *virtuellen Servern* verknüpft ist. Ein virtueller Server ist ein Dienst, der zum Lauschen auf einer bestimmten virtuellen IP konfiguriert ist. In Abschnitt 4.6, „**VIRTUAL SERVERS**“ finden Sie weitere Informationen über die Konfiguration eines virtuellen Servers mithilfe des **Piranha-Konfigurationstools**. Eine VIP-Adresse migriert im Falle eines Ausfalls von einem LVS-Router auf den anderen und gewährleistet somit die Verfügbarkeit an der IP-Adresse (auch *Floating-IP-Adresse* genannt).

VIP-Adressen können eine Alias-Bezeichnung für dasselbe Gerät erhalten, das den LVS-Router mit dem Internet verbindet. Wenn zum Beispiel eth0 mit dem Internet verbunden ist, dann können mehrere virtuelle Server eine Alias-Bezeichnung **`eth0:1`** erhalten. Alternativ kann jeder virtuelle Server mit einem separaten Gerät pro Dienst verknüpft werden. Beispielsweise kann HTTP-Datenverkehr auf **`eth0:1`** und FTP-Datenverkehr auf **`eth0:2`** verwaltet werden.

Nur jeweils ein LVS-Router ist aktiv. Die Aufgabe des aktiven Routers ist es, Dienstanfragen von virtuellen IP-Adressen an die realen Server umzuleiten. Die Umleitung basiert auf einem von acht Algorithmen für die Lastverteilung, die in Abschnitt 1.3, „Übersicht über das Load Balancer Add-On Scheduling“ näher beschrieben werden.

Der aktive Router überwacht zudem dynamisch die Gesamtverfassung der speziellen Dienste auf den realen Servern durch einfache *send/expect-Skripte*. Als Hilfe bei der Analyse der Verfassung eines Dienstes, der dynamische Daten wie HTTPS oder SSL benötigt, kann der Administrator auch externe ausführbare Dateien aufrufen. Falls ein Dienst auf einem realen Server nicht ordnungsgemäß funktioniert, hört der aktive Router auf, Jobs an diesen Server zu senden, bis dieser wieder ordnungsgemäß läuft.

Der Backup-Router übernimmt die Rolle eines Systems in Bereitschaft. Der LVS-Router tauscht regelmäßig Heartbeat-Meldungen über die primäre externe öffentliche Schnittstelle und im Falle eines Failovers über die private Schnittstelle aus. Erhält der Backup-Router keine Heartbeat-Meldung innerhalb eines erwarteten Intervalls, leitet er die Ausfallsicherung ein und übernimmt die Rolle des aktiven Routers. Während der Ausfallsicherung übernimmt der Backup-Router die VIP-Adressen, die vom ausgefallenen Router bereitgestellt wurden, unter Verwendung einer Technik, die als *ARP-Spoofing* bekannt ist – hierbei zeigt der Backup-LVS-Router an, dass er das Ziel für IP-Adressen darstellt, die an den ausgefallenen Knoten gerichtet sind. Falls der ausgefallene Knoten wieder aktiv wird, nimmt der Backup-Knoten seine Backup-Rolle wieder auf.

Die einfache Konfiguration mit zwei Schichten in Abbildung 1.1, „Eine grundlegende Load Balancer Add-On Konfiguration“ ist am besten geeignet für die Bereitstellung von Daten, die sich nicht sehr häufig ändern – wie z. B. eine statische Website – da die einzelnen realen Server nicht automatisch die Daten zwischen den Knoten synchronisieren.

1.1.1. Daten replizieren und gemeinsam verwenden auf realen Servern

Da es keine integrierte Komponente im Load Balancer Add-On gibt, die dieselben Daten auf den realen Servern verteilt, hat der Administrator zwei Möglichkeiten:

- Synchronisation der Daten auf den realen Servern
- Hinzufügen einer dritten Schicht zur Topologie für den Zugriff auf gemeinsam genutzte Daten

Die erste Option wird vorzugsweise für Server verwendet, die einer großen Anzahl von Benutzern das Hochladen oder Verändern von Daten auf den realen Servern untersagt. Falls die Konfiguration es einer großen Anzahl von Benutzern gestattet, Daten zu verändern, wie beispielsweise einer E-Commerce-Website, ist das Hinzufügen einer dritten Schicht besser.

1.1.1.1. Konfigurieren von realen Servern zur Synchronisierung von Daten

Es gibt viele Möglichkeiten, wie ein Administrator Daten im Pool der realen Server synchronisieren kann. Beispielsweise können Shell-Skripte verwendet werden, die, sobald eine Website aktualisiert wird, diese Seite an alle Server gleichzeitig verteilen. Der Systemadministrator kann auch Programme wie **rsync** dazu verwenden, um veränderte Daten in bestimmten Abständen auf alle Knoten zu replizieren.

Allerdings funktioniert diese Art der Datensynchronisation nicht optimal, falls die Konfiguration überladen ist mit Benutzern, die permanent Dateien hochladen oder Datenbanktransaktionen vornehmen. Für eine Konfiguration mit hoher Auslastung ist eine *dreischichtige Topologie* die beste Lösung.

1.2. Eine dreischichtige Load Balancer Add-On Konfiguration

Abbildung 1.2, „Eine dreischichtige Load Balancer Add-On Konfiguration“ zeigt eine typische dreischichtige Load Balancer Add-On Topologie. In diesem Beispiel leitet der aktive LVS-Router die Anfragen vom Internet an den Pool der realen Server. Jeder dieser realen Server greift dann auf eine gemeinsame Datenquelle auf dem Netzwerk zu.

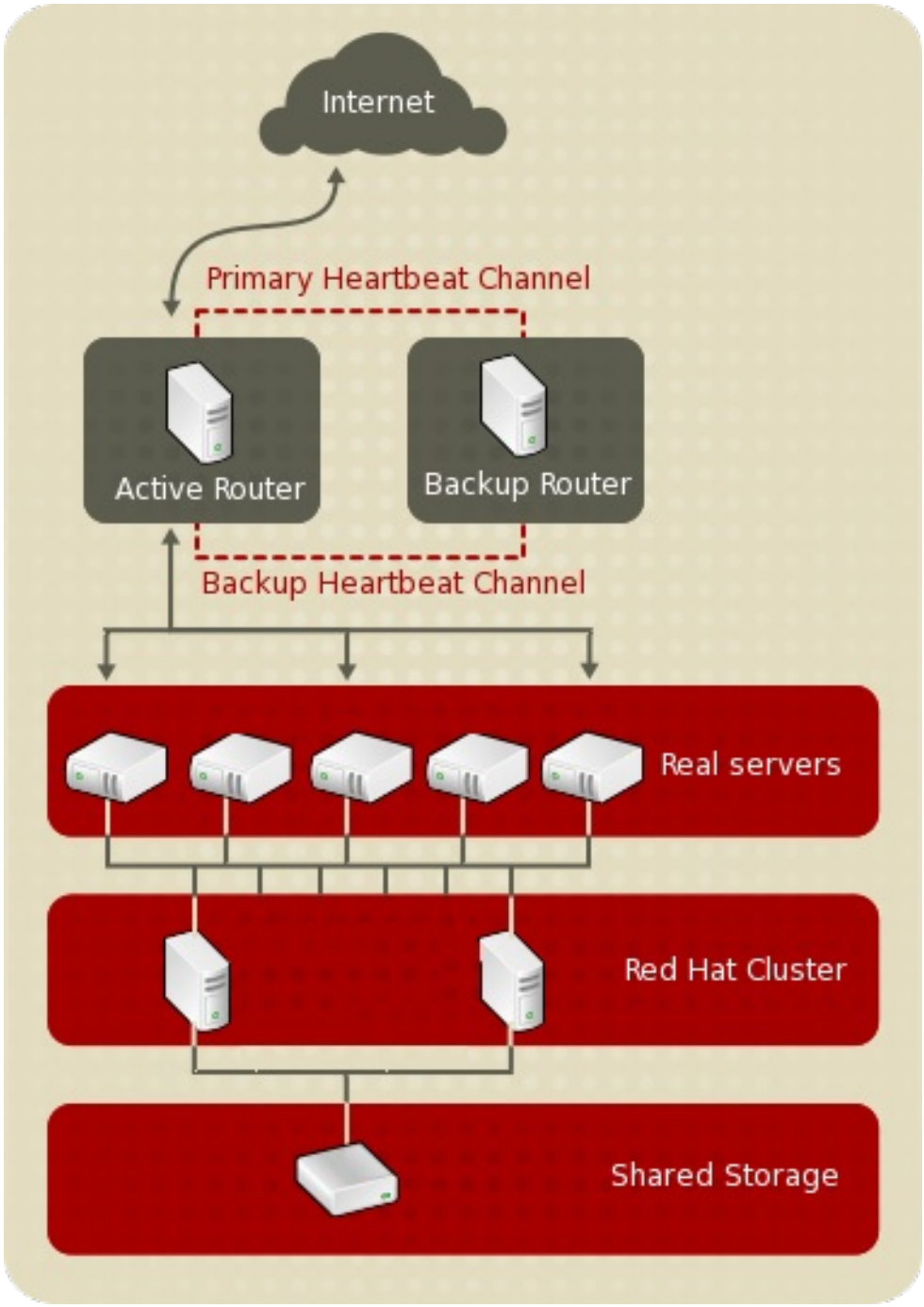


Abbildung 1.2. Eine dreischichtige Load Balancer Add-On Konfiguration

Diese Konfiguration ist ideal für gut ausgelastete FTP-Server, bei denen Daten, auf die zugegriffen werden kann, auf einem hochverfügbaren Server gespeichert werden und auf die von jedem realen Server über ein exportiertes NFS-Verzeichnis oder eine Samba-Freigabe zugegriffen wird. Diese Topologie wird außerdem für Websites empfohlen, die auf eine zentrale, hochverfügbare Datenbank für Transaktionen zugreifen. Zusätzlich können Administratoren unter Verwendung einer aktiv-aktiv-Konfiguration für das Load Balancer Add-On ein Hochverfügbarkeits-Cluster so konfigurieren, dass er diese beiden Rollen gleichzeitig ausübt.

Die dritte Schicht in dem obigen Beispiel muss nicht unbedingt das Load Balancer Add-on verwenden, allerdings bedeutet das Fehlen einer hochverfügbaren Lösung, dass es hier einen Single Point of Failure gibt.

1.3. Übersicht über das Load Balancer Add-On Scheduling

Einer der Vorteile bei der Verwendung des Load Balancer Add-Ons ist dessen Fähigkeit zur flexiblen, IP-basierten Lastverteilung auf dem Pool der realen Server. Diese Flexibilität verdankt es der Vielzahl an Scheduling-Algorithmen, aus denen ein Administrator bei der Konfiguration des Load Balancer Add-Ons auswählen kann. Die Lastverteilung mithilfe des Load Balancer Add-Ons ist weniger flexiblen Methoden wie z. B. *Round-Robin DNS* überlegen, bei denen das hierarchische Wesen von DNS und das Caching durch Client-Rechner zu unausgeglichener Last führen kann. Darüber hinaus bietet die Low-Level-Filterung durch den LVSR-Router Vorteile gegenüber der Anfrageweiterleitung auf Applikationsebene, da die Lastverteilung auf Netzwerkpaket-Ebene nur einen geringen Overhead verursacht und eine größere Skalierbarkeit ermöglicht.

Beim Scheduling kann der aktive Router die Aktivität der realen Server sowie optional die vom Administrator zugewiesene *Gewichtung* für das Routing von Dienstanfragen berücksichtigen. Diese zugewiesene Gewichtung gibt einzelnen Rechnern eine beliebige Priorität. Bei dieser Form des Schedulings ist es möglich, aus einer Vielzahl verschiedener Hardware- und Software-Kombinationen eine Gruppe aus Servern zu bilden, auf denen der aktive Router die Last gleichmäßig verteilen kann.

Der Scheduling-Mechanismus für das Load Balancer Add-On wird von einer Reihe von Kernel-Patches namens *IP Virtual Server* oder *IPVS*-Modulen bereitgestellt. Diese Module ermöglichen das *Layer 4* (*L4*) Switching der Transportschicht, was für den Einsatz auf mehreren Servern mit einer einzigen IP-Adresse optimiert ist.

Um Pakete wirkungsvoll weiterzuleiten und nachzuverfolgen, erstellt IPVS eine *IPVS-Tabelle* im Kernel. Diese Tabelle wird vom aktiven LVS-Router verwendet, um Anfragen von einer virtuellen Serveradresse und Antworten von den realen Servern im Pool weiterzuleiten. Die IPVS-Tabelle wird laufend aktualisiert von einem Dienstprogramm namens *ipvsadm*, das Cluster-Mitglieder abhängig von deren Verfügbarkeit hinzufügt oder entfernt.

1.3.1. Scheduling-Algorithmen

Die Struktur der IPVS-Tabelle hängt vom Scheduling-Algorithmus ab, den der Administrator für den virtuellen Server festlegt. Red Hat Enterprise Linux bietet die nachfolgend aufgeführten Scheduling-Algorithmen und bietet so ein Höchstmaß an Flexibilität für die Arten von Diensten, die geclustert werden können, sowie für die Art und Weise, wie das Scheduling für diese Dienste erfolgt. Anweisungen zur Zuweisung von Scheduling-Algorithmen finden Sie in Abschnitt 4.6.1, „Der Unterabschnitt **VIRTUAL SERVER**“.

Round-Robin-Scheduling

Verteilt die Anfragen reihum in dem Pool der realen Server. Bei der Verwendung dieses Algorithmus werden alle realen Server als gleichwertig behandelt, ohne Rücksicht auf deren Kapazität oder Auslastung. Dieses Scheduling-Modell ähnelt Round-Robin-DNS, ist jedoch feiner steuerbar, da es auf Netzwerkverbindungen basiert statt auf Hosts. Das Round-Robin-Scheduling des Load Balancer Add-Ons leidet zudem nicht unter dem Ungleichgewicht, das gecachte DNS-Anfragen verursachen.

Gewichtetes Round-Robin-Scheduling

Verteilt die Anfragen reihum in dem Pool der realen Server, teilt jedoch Servern mit einer höheren Kapazität mehr Jobs zu. Die Kapazität wird durch einen vom Administrator zugewiesenen Gewichtungsfaktor angezeigt, der dann durch eine dynamische Auslastungsinformation entweder nach oben oder unten korrigiert wird. Siehe Abschnitt 1.3.2, „Server-Gewichtung und Scheduling“ für weitere Informationen über die Gewichtung der realen Server.

Das gewichtete Round-Robin-Scheduling ist die bevorzugte Methode, falls es deutliche Unterschiede in der Kapazität der realen Server im Pool gibt. Falls die Anfragelast jedoch sehr schwankt, bedient der höher gewichtete Server unter Umständen einen zu großen Teil

der Anfragen.

Least-Connection

Verteilt mehr Anfragen an reale Server mit weniger aktiven Verbindungen. Da diese Methode anhand der IPVS-Tabelle die Anzahl der aktiven Verbindungen zu den realen Servern nachverfolgt, ist dies ein dynamischer Scheduling-Algorithmus-Typ und stellt eine bessere Wahl dar, falls die Anfragelast sehr schwankt. Dieser Algorithmus ist am besten für einen Pool von realen Server geeignet, in dem jeder Mitgliedsknoten ungefähr dieselbe Kapazität besitzt. Falls die realen Server unterschiedliche Kapazitäten besitzen, ist das gewichtete Least-Connection-Scheduling die bessere Wahl.

Gewichtetes Least-Connections (Standard)

Verteilt mehr Anfragen an Server mit weniger aktiven Verbindungen in Relation zu ihrer Kapazität. Die Kapazität wird durch eine benutzerdefinierte Gewichtung angezeigt, welche dann wiederum je nach dynamischer Lastinformation nach oben oder unten korrigiert wird. Der Zusatz der Gewichtung macht diesen Algorithmus zu einer idealen Alternative, wenn der Pool der realen Server aus Hardware mit unterschiedlichen Kapazitäten besteht. Siehe Abschnitt 1.3.2, „Server-Gewichtung und Scheduling“ für weitere Informationen über die Gewichtung der realen Server.

Standortbasiertes Least-Connection-Scheduling

Verteilt mehr Anfragen an Server mit weniger aktiven Verbindungen in Relation zu ihren Ziel-IPs. Dieser Algorithmus ist für den Einsatz in einem Proxy-Cache-Server-Cluster konzipiert. Er leitet die Pakete für eine IP-Adresse solange an den Server für diese Adresse, bis dieser Server seine Kapazitätsgrenze überschreitet und einen Server mit dessen halber Last vorliegen hat. In diesem Fall weist er die IP-Adresse dem realen Server mit der geringsten Auslastung zu.

Standortbasiertes Least-Connection-Scheduling mit Replikations-Scheduling

Verteilt mehr Anfragen an Server mit weniger aktiven Verbindungen in Relation zu ihren Ziel-IPs. Dieser Algorithmus ist ebenfalls für den Einsatz in einem Proxy-Cache-Server-Cluster konzipiert. Er unterscheidet sich vom standortbasierten Least-Connection-Scheduling durch das Zuordnen der Ziel-IP-Adresse zu einer Teilmenge von realen Server-Knoten. Anfragen werden anschließend zum Server in dieser Teilmenge mit der niedrigsten Anzahl an Verbindungen geroutet. Falls alle Knoten für die Ziel-IP über der Kapazität liegen, repliziert er einen neuen Server für diese Ziel-IP-Adresse, indem er den realen Server mit den wenigsten Verbindungen aus dem gesamten Pool der realen Server zur Teilmenge der realen Server für diese Ziel-IP hinzufügt. Der Knoten mit der höchsten Last wird dann aus der Teilmenge der realen Server entfernt, um eine Überreplikation zu verhindern.

Ziel-Hash-Scheduling

Verteilt Anfragen an den Pool realer Server, indem die Ziel-IP in einer statischen Hash-Tabelle gesucht wird. Dieser Algorithmus ist für einen Proxy-Cache-Server-Cluster konzipiert.

Quell-Hash-Scheduling

Verteilt Anfragen an den Pool realer Server, indem die Quell-IP in einer statischen Hash-Tabelle gesucht wird. Dieser Algorithmus ist für LVS-Router mit mehreren Firewalls konzipiert.

1.3.2. Server-Gewichtung und Scheduling

Der Administrator des Load Balancer Add-Ons kann jedem Knoten im Pool der realen Server eine *Gewichtung* zuordnen. Diese Gewichtung ist ein ganzzahliger Wert, der bei jedem *gewichtungsabhängigen* Scheduling-Algorithmus berücksichtigt wird (beispielsweise beim gewichteten Least-Connections) und die dem LVS-Router dabei hilft, Hardware mit unterschiedlicher Kapazität gleichmäßig auszulasten.

Die Gewichtung bestimmt das relative Verhältnis untereinander. Falls ein Server beispielsweise die Gewichtung 1 hat und ein anderer die Gewichtung 5, dann erhält der Server mit der Gewichtung 5 fünfmal so viele Verbindungen wie der Server mit der Gewichtung 1. Der Standardwert für die Gewichtung eines realen Servers ist 1.

Das Hinzufügen einer Gewichtung zu verschiedenen Hardware-Konfigurationen in einem realen Server-Pool kann zwar eine effizientere Lastverteilung begünstigen, allerdings kann es auch zu einem vorübergehenden Ungleichgewicht führen, wenn ein realer Server zum Pool der realen Server hinzugefügt wird und der virtuelle Server das gewichtete Least-Connections-Scheduling verwendet. Angenommen, es gibt drei Server im realen Server-Pool. Server A und B sind mit 1 gewichtet, während Server C mit 2 gewichtet ist. Falls Server C ausfällt, wird die fallengelassene Last gleichmäßig auf die verbleibenden Server A und B verteilt. Sobald der Server C jedoch wieder online ist, sieht der LVS-Router, dass dieser keinerlei Verbindungen hat und überflutet Server C mit allen eingehenden Anfragen, bis dieser auf dem gleichen Level wie Server A und B ist.

Um dieses Phänomen zu verhindern, können Administratoren den virtuellen Server als *Quiesce*-Server definieren, sodass der reale Server C im obigen Beispiel nicht aus der Tabelle des virtuellen Servers entfernt wird. Stattdessen wird seine Gewichtung auf 0 gesetzt, wodurch er effektiv deaktiviert wird. Sobald der reale Server C wieder verfügbar wird, erhält er seine ursprüngliche Gewichtung wieder und wird so wieder aktiviert.

1.4. Routing-Methoden

Red Hat Enterprise Linux verwendet *Network Address Translation* oder kurz *NAT-Routing* für das Load Balancer Add-On, was dem Administrator eine enorme Flexibilität beim Einsatz der verfügbaren Hardware und bei der Integration des Load Balancer Add-Ons in ein vorhandenes Netzwerk ermöglicht.

1.4.1. NAT-Routing

Abbildung 1.3, „Load Balancer Add-On implementiert mit NAT-Routing“ veranschaulicht, wie das Load Balancer Add-On das NAT-Routing zur Weiterleitung von Anfragen zwischen dem Internet und einem privaten Netzwerk verwendet.

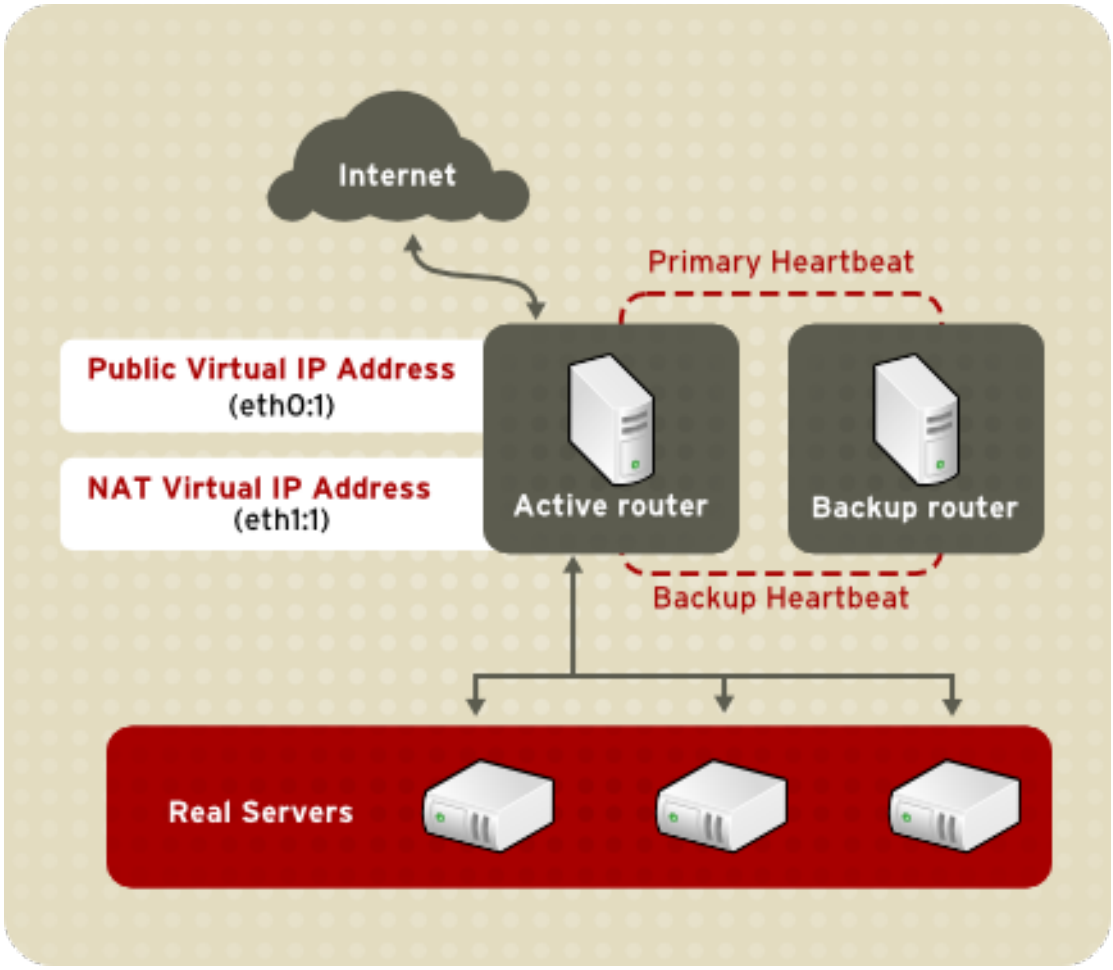


Abbildung 1.3. Load Balancer Add-On implementiert mit NAT-Routing

In diesem Beispiel verfügt der aktive LVS-Router über zwei Netzwerkkarten (NICs). Die NIC für das Internet besitzt eine *reale IP-Adresse* auf eth0 sowie eine Floating-IP-Adresse für einen Alias auf eth0:1. Die NIC für die private Netzwerkschnittstelle besitzt eine reale IP-Adresse auf eth1 sowie eine Floating-IP-Adresse für einen Alias auf eth1:1. Bei einer Ausfallsicherung wird die virtuelle Netzwerkschnittstelle, die sich zum Internet hin richtet, sowie die nach privat ausgerichtete virtuelle Schnittstelle gleichzeitig vom Backup-LVS-Router übernommen. Alle realen Server im privaten Netzwerk verwenden die Floating-IP-Adresse für den NAT-Router als ihre standardmäßige Route, um mit dem aktiven LVS-Router zu kommunizieren, sodass ihre Fähigkeiten, auf Anfragen aus dem Internet zu reagieren, nicht beeinträchtigt werden.

In diesem Beispiel werden für die öffentliche Floating-LVS-IP-Adresse des LVS-Routers und die private Floating-NAT-IP-Adresse zwei Aliasse für die physischen NICs erstellt. Auch wenn es möglich ist, jede Floating-IP-Adresse mit ihrem physischen Gerät auf den LVS-Router-Knoten zu verknüpfen, ist das Vorhandensein von mehr als zwei NICs keine Voraussetzung.

Bei der Verwendung dieser Topologie erhält der aktive LVS-Router die Anfragen und routet sie an den entsprechenden Server weiter. Der reale Server verarbeitet anschließend die Anfrage und gibt die Pakete an den LVS-Router zurück. Der LVS-Router verwendet Network Address Translation, um die Adresse des realen Servers in den Paketen durch die öffentliche VIP-Adresse der LVS-Router zu ersetzen. Dieser Prozess wird als *IP-Masquerading* bezeichnet, da die tatsächlichen IP-Adressen der realen Server vor den anfragenden Clients verborgen bleibt.

Beim Einsatz von NAT-Routing kann es sich bei den realen Servern um beliebige Rechner handeln, auf denen verschiedene Betriebssysteme laufen. Der Hauptnachteil ist, dass der LVS-Router in größeren Cluster-Bereitstellungen zu einem Engpass werden kann, da er ausgehende und eingehende Anfragen verarbeiten muss.

1.4.2. Direktes Routing

Das Einrichten einer Load Balancer Add-On Konfiguration, die direktes Routing verwendet, bietet den Vorteil einer höheren Leistung im Vergleich zu anderen Netzwerktopologien für das Load Balancer Add-On. Direktes Routing ermöglicht den realen Servern, Pakete direkt zu verarbeiten und an einen anfragenden Benutzer zu routen, anstatt alle ausgehende Pakete durch den LVS-Router zu leiten. Direktes Routing vermindert die Wahrscheinlichkeit von Problemen bei der Netzwerkleistung, indem es den Job des LVS-Routers nur auf das Verarbeiten von eingehenden Paketen beschränkt.

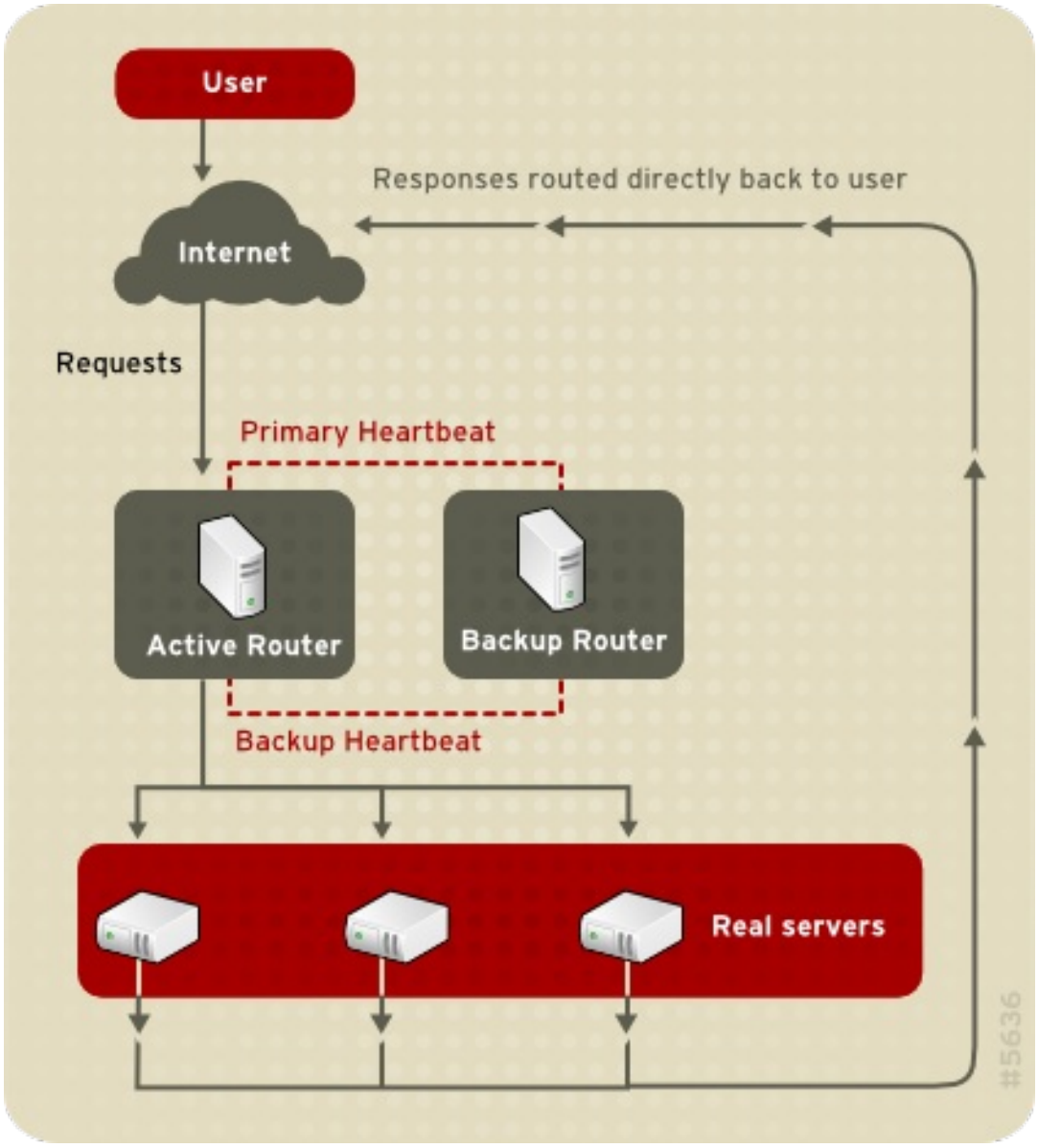


Abbildung 1.4. Load Balancer Add-On implementiert mit direktem Routing

In einer typischen Load Balancer Add-On Konfiguration mit direktem Routing empfängt ein eingehender Server Anfragen über die virtuelle IP (VIP) und verwendet einen Scheduling-Algorithmus, um die Anfragen an die realen Server zu routen. Der reale Server verarbeitet die Anfrage und sendet die Antwort direkt an den Client und umgeht so die LVS-Router. Direktes Routing ermöglicht Skalierbarkeit, sodass reale Server hinzugefügt werden können, ohne dass der LVS-Router zusätzlich damit belastet wird, ausgehende Pakete vom realen Server zum Client zu routen, was bei hoher Netzwerkauslastung zu einem Engpass führen kann.

1.4.2.1. Direktes Routing und die ARP-Einschränkung

Auch wenn es bei der Verwendung von direktem Routing im Load Balancer Add-On viele Vorteile gibt, so existieren doch einige Einschränkungen. Das häufigste Problem mit direktem Routing im Load Balancer Add-On tritt beim *Address Resolution Protocol* (ARP) auf.

In typischen Situationen sendet ein Client aus dem Internet eine Anfrage an eine IP-Adresse. Netzwerk-Router senden Anfragen normalerweise an ihr Ziel, indem sie IP-Adressen mittels ARP mit einer MAC-Adresse eines Rechners in Beziehung bringen. ARP-Anfragen werden an alle im Netzwerk verbundenen Rechner verbreitet und der Rechner mit der korrekten IP-/MAC-Adresskombination erhält das Paket. Die IP-/MAC-Verbindungen werden in einem ARP-Cache gespeichert, welcher regelmäßig gelöscht (normalerweise alle 15 Minuten) und neu mit IP-/MAC-Verbindungen gefüllt wird.

Das Problem mit ARP-Anfragen in einer Load Balancer Add-On Konfiguration mit direktem Routing ist, dass aufgrund der Tatsache, dass eine Client-Anfrage an eine IP-Adresse mit einer MAC-Adresse verknüpft werden muss, damit diese Anfrage bearbeitet werden kann, die virtuelle IP-Adresse des Load Balancer Add-On Systems ebenfalls mit einer MAC verknüpft sein muss. Da aber sowohl der LVS-Router als auch die realen Server alle dieselbe VIP besitzen, wird die ARP-Anfrage an alle mit dem VIP verknüpften Rechner verbreitet. Dies kann zu einigen Problemen führen, z. B. dass die VIP direkt mit einem der realen Server verknüpft wird und Anfragen direkt weiterleitet und dabei den LVS-Router komplett umgeht, was dem Zweck der Load Balancer Add-On Konfiguration zuwider läuft.

Um dieses Problem zu lösen, vergewissern Sie sich, dass die eingehenden Anfragen immer an den LVS-Router gesendet werden statt an einen der realen Server. Sie erreichen dies, indem Sie entweder das Paketfilterungstool **`arptables_jf`** oder **`iptables`** verwenden, denn:

- **`arptables_jf`** hindert ARP an der Verknüpfung von VIPs mit realen Servern.
- Die **`iptables`**-Methode umgeht das ARP-Problem, indem VIPs gar nicht erst auf realen Servern konfiguriert werden.

Weitere Informationen über die Verwendung von **`arptables`** oder **`iptables`** in einer Load Balancer Add-On Umgebung mit direktem Routing finden Sie in Abschnitt 3.2.1, „Direktes Routing und **`arptables_jf`**“ oder Abschnitt 3.2.2, „Direktes Routing und **`iptables`**“.

1.5. Persistenz und Firewall-Markierungen

In bestimmten Situationen kann es wünschenswert sein, dass sich ein Client immer wieder mit demselben realen Server verbindet, statt diese Anfrage durch einen Lastverteilungs-Algorithmus des Load Balancer Add Ons an einen anderen verfügbaren Server zu schicken. Beispiele für eine solche Situation umfassen Web-Formulare, die sich über mehrere Bildschirme erstrecken, Cookies, SSL- und FTP-Verbindungen. In diesen Fällen funktioniert ein Client ggf. nicht ordnungsgemäß, wenn die Transaktionen nicht von demselben Server gehandhabt wird, um den Kontext beizubehalten. Das Load Balancer Add-On bietet zwei verschiedene Features, um dies zu handhaben: *Persistenz* und *Firewall-Markierungen*.

1.5.1. Persistenz

Wenn die Persistenz aktiviert ist, fungiert sie wie ein Timer. Wenn sich ein Client mit einem Dienst verbindet, merkt sich das Load Balancer Add-On die letzte Verbindung für eine festgelegte Zeitspanne. Falls sich dieselbe Client-IP-Adresse innerhalb dieser Zeitspanne erneut verbindet, wird sie an denselben Server weitergeleitet, mit dem sie zuvor bereits verbunden wurde – dabei werden die Mechanismen zum Lastverteilung übergangen. Falls eine Verbindung außerhalb des Zeitfensters zustande kommt, wird sie anhand der derzeit aktiven Scheduling-Regeln gehandhabt.

Persistenz erlaubt dem Administrator auch die Angabe einer Subnetzmaske, die für einen Client-IP-Adresstest angewendet werden kann, als ein Tool zur Kontrolle, welche Adressen ein höheres Level an Persistenz besitzen, sodass Verbindungen mit diesem Subnetz gruppiert werden.

Das Gruppieren von Verbindungen, die für verschiedene Ports bestimmt sind, kann für Protokolle von Bedeutung sein, die mehr als einen Port für die Kommunikation verwenden, wie beispielsweise FTP. Allerdings stellt Persistenz nicht die effektivste Art und Weise dar, Probleme mit der Gruppierung von Verbindungen, die für verschiedene Ports bestimmt sind, zu lösen. In diesen Situationen sollten am besten *Firewall-Markierungen* verwendet werden.

1.5.2. Firewall-Markierungen

Firewall-Markierungen sind eine einfache und effektive Art und Weise, um Ports zu gruppieren, die für ein Protokoll oder eine Gruppe von verwandten Protokollen verwendet werden. Wenn das Load Balancer Add-On beispielsweise im Rahmen einer E-Commerce-Website eingesetzt wird, können Firewall-Markierungen dazu verwendet werden, um HTTP-Verbindungen auf Port 80 und sichere HTTPS-Verbindungen auf Port 443 zu bündeln. Indem für jedes dieser Protokolle jeweils dieselbe Firewall-Markierung zum virtuellen Server zugewiesen wird, können Informationen über den Zustand der Transaktion erhalten bleiben, da der LVS-Router alle Anfragen an denselben realen Server weiterleitet, nachdem eine Verbindung geöffnet wurde.

Aufgrund seiner Effizienz und der einfachen Handhabung sollten Administratoren des Load Balancer Add-Ons wann immer möglich Firewall-Markierungen statt Persistenz zum Gruppieren von Verbindungen verwenden. Allerdings sollten Administratoren nach wie vor Persistenz in Verbindung mit Firewall-Markierungen zu den virtuellen Servern hinzufügen, um sicherzustellen, dass Clients während einer angemessenen Zeitspanne wieder mit demselben Server verbunden werden.

1.6. Load Balancer Add-On – Ein Blockdiagramm

LVS-Router verwenden eine Reihe von Programmen zu Überwachung der Cluster-Mitglieder und Cluster-Dienste. Abbildung 1.5, „Load Balancer Add-On Komponenten“ veranschaulicht, wie diese verschiedenen Programme auf dem aktiven und dem Backup-LVS-Router zusammenarbeiten, um den Cluster zu verwalten.

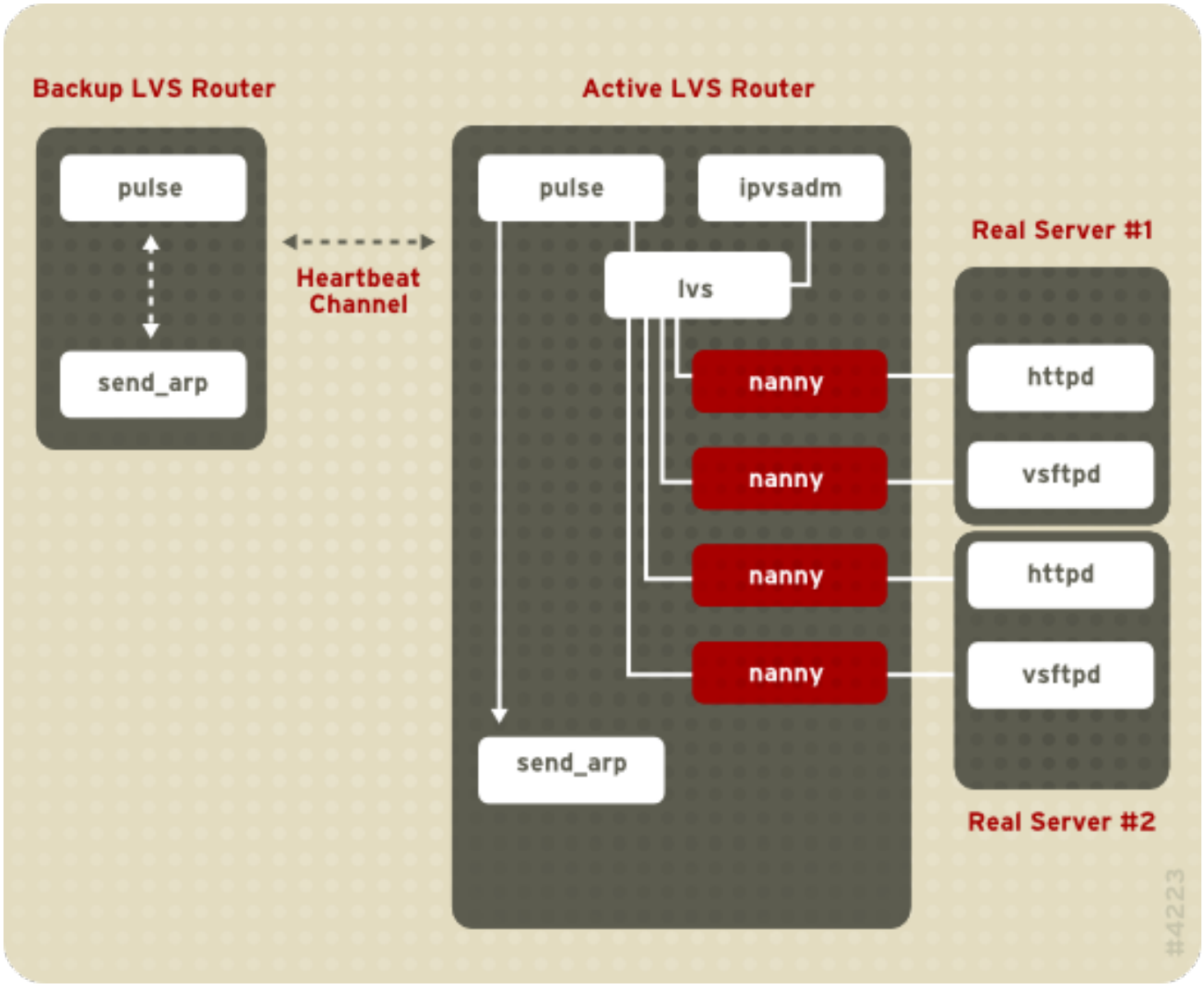


Abbildung 1.5. Load Balancer Add-On Komponenten

Der **pulse**-Daemon läuft sowohl auf den aktiven als auch den passiven LVS-Routern. Auf dem Backup-Router sendet **pulse** einen *Heartbeat* an die öffentliche Schnittstelle des aktiven Routers, um sicherzustellen, dass der aktive Router noch ordnungsgemäß funktioniert. Auf dem aktiven Router startet **pulse** den **lvs**-Daemon und antwortet auf *Heartbeat*-Anfragen des Backup-LVS-Routers.

Sobald er gestartet wird, ruft der **lvs**-Daemon das **ipvsadm**-Dienstprogramm auf, um die IPVS-Routing-Tabelle im Kernel zu konfigurieren und zu pflegen, und startet einen **nanny**-Prozess für jeden konfigurierten virtuellen Server auf jedem realen Server. Jeder **nanny**-Prozess überprüft den Status eines konfigurierten Dienstes auf einem realen Server und unterrichtet den **lvs**-Daemon darüber, ob der Dienst auf diesem realen Server korrekt funktioniert. Falls eine Fehlfunktion entdeckt wird, weist der **lvs**-Daemon **ipvsadm** an, diesen realen Server aus der IPVS-Routing-Tabelle zu entfernen.

Falls der Backup-Router keine Antwort vom aktiven Router erhält, leitet er einen Failover ein, indem er **send_arp** aufruft, um alle virtuellen IP-Adressen den NIC-Hardware-Adressen (*MAC* Adressen) des Backup-Knotens neu zuzuweisen. Weiterhin sendet er einen Befehl an den aktiven Router sowohl über die öffentliche als auch die private Netzwerkschnittstelle, den **lvs**-Daemon auf dem aktiven Router zu beenden, und startet den **lvs**-Daemon auf dem Backup-Router, um Anfragen für die konfigurierten virtuellen Server zu akzeptieren.

1.6.1. Load Balancer Add-On Komponenten

Abschnitt 1.6.1.1, „**pulse**“ zeigt eine detaillierte Liste aller Softwarekomponenten in einem LVS-Router.

1.6.1.1. pulse

Dies ist der Steuerprozess, der alle anderen Daemons startet, die mit den LVS-Routern in Verbindung stehen. Zum Zeitpunkt des Bootens wird der Daemon mithilfe des Skripts **/etc/rc.d/init.d/pulse** gestartet. Der Daemon liest dann die Konfigurationsdatei **/etc/sysconfig/ha/lvs.cf**. Auf dem aktiven Router startet **pulse** den LVS-Daemon. Auf dem Backup-Router bestimmt **pulse** die Verfassung des aktiven Routers, indem es einen einfachen Heartbeat in benutzerdefinierten Abständen ausgibt. Falls der aktive Router sich nicht nach einem benutzerdefinierten Intervall meldet, wird ein Failover eingeleitet. Während des Failovers weist **pulse** auf dem Backup-Router den **pulse**-Daemon auf dem aktiven Router an, alle LVS-Dienste herunterzufahren, startet das Programm **send_arp**, um die relativen IP-Adressen der MAC-Adresse des Backup-Routers neu zuzuweisen und startet den **lvs**-Daemon.

1.6.1.2. lvs

Der **lvs**-Daemon läuft auf dem aktiven LVS-Router, sobald er von **pulse** aufgerufen wird. Er liest die Konfigurationsdatei **/etc/sysconfig/ha/lvs.cf**, ruft das Dienstprogramm **ipvsadm** auf, um die IPVS-Routing-Tabelle zu erstellen und zu pflegen und weist jedem konfigurierten Load Balancer Add-On Dienst einen **nanny**-Prozess zu. Falls der **nanny**-Prozess meldet, dass ein realer Server nicht mehr erreichbar ist, weist **lvs** das Dienstprogramm **ipvsadm** an, den realen Server aus der IPVS-Routing-Tabelle zu entfernen.

1.6.1.3. ipvsadm

Dieser Dienst aktualisiert die IPVS-Routing-Tabelle im Kernel. Der **lvs**-Daemon richtet das Load Balancer Add-On ein und verwaltet es, indem er **ipvsadm** aufruft, um Einträge in der IPVS-Routing-Tabelle hinzuzufügen, zu ändern oder zu löschen.

1.6.1.4. nanny

Der Überwachungs-Daemon **nanny** läuft auf dem aktiven LVS-Router. Mithilfe dieses Daemons bestimmt der aktive Router die Verfassung eines jeden realen Servers und überwacht optional dessen Auslastung. Ein separater Prozess läuft für jeden Dienst, der auf jedem realen Server definiert ist.

1.6.1.5. /etc/sysconfig/ha/lvs.cf

Dies ist die Konfigurationsdatei des Load Balancer Add-Ons. Alle Daemons erhalten ihre Konfigurationsinformationen direkt oder indirekt von dieser Datei.

1.6.1.6. Piranha-Konfigurationstool

Dies ist das webbasierte Tool zur Überwachung, Konfiguration und Administration des Load Balancer Add-Ons. Es ist das Standardtool zur Pflege der Load Balancer Add-On Konfigurationsdatei **/etc/sysconfig/ha/lvs.cf**.

1.6.1.7. send_arp

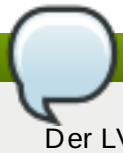
Dieses Programm sendet ARP-Broadcasts, wenn die Floating-IP-Adresse während des Failovers von einem Knoten auf einen anderen wechselt.

Kapitel 2, *Erste Konfigurationsschritte für das Load Balancer Add-On* behandelt wichtige

Konfigurationsschritte nach erfolgter Installation, die Sie durchführen sollten, bevor Sie Red Hat Enterprise Linux zum Einsatz als LVS-Router konfigurieren.

Kapitel 2. Erste Konfigurationsschritte für das Load Balancer Add-On

Nach der Installation von Red Hat Enterprise Linux müssen Sie einige grundlegende Konfigurationsschritte durchführen, um den LVS-Router und die Server zu konfigurieren. In diesem Kapitel werden diese ersten Schritte detailliert erläutert.

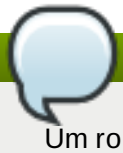


Anmerkung

Der LVS-Routerknoten, der als aktiver Knoten dient, sobald das Load Balancer Add-On läuft, wird auch als *primärer Knoten* bezeichnet. Verwenden Sie zur Konfiguration des Load Balancer Add-Ons das **Piranha-Konfigurationstool** auf dem primären Knoten.

2.1. Konfigurieren von Diensten auf dem LVS-Router

Das Red Hat Enterprise Linux Installationsprogramm installiert alle Komponenten, die zum Einrichten des Load Balancer Add-Ons notwendig sind. Allerdings müssen vor der Konfiguration des Load Balancer Add-Ons die richtigen Dienste aktiviert werden. Für den LVS-Router müssen beim Systemstart die richtigen Dienste starten. Unter Red Hat Enterprise Linux gibt es drei wesentliche Werkzeuge, mit denen Sie Dienste beim Systemstart aktivieren können: das Befehlszeilenprogramm **chkconfig**, das ncurses-basierte Programm **ntsysv** und das grafische **Dienst-Konfigurationstool**. Alle drei erfordern root-Zugriff.



Anmerkung

Um root-Zugriff zu erhalten, öffnen Sie eine Shell-Eingabeaufforderung und geben Sie den Befehl **su -** gefolgt vom root-Passwort ein. Zum Beispiel:

```
$ su - root password
```

Auf dem LVS-Router gibt es drei Dienste, die zur Aktivierung beim Systemstart konfiguriert werden müssen:

- Der **piranha-gui**-Dienst (nur auf dem primären Knoten)
- Der **pulse**-Dienst
- Der **sshd**-Dienst

Falls Sie Multi-Port-Dienste clustern oder Firewall-Markierungen verwenden, müssen Sie zudem den **iptables**-Dienst aktivieren.

Es empfiehlt sich, diese Dienste sowohl in Runlevel 3 als auch in Runlevel 5 zu aktivieren. Führen Sie dazu den folgenden **chkconfig**-Befehl für jeden Dienst aus:

```
/sbin/chkconfig --level 35 daemon on
```

Ersetzen Sie in dem obigen Befehl *daemon* durch den Namen des Dienstes, den Sie aktivieren. Mithilfe des folgenden Befehls erhalten Sie eine Liste von Diensten auf dem System sowie Angaben darüber, auf welchen Runlevels diese Dienste aktiviert sind:

```
/sbin/chkconfig --list
```

Warnung

Wenn Sie die oben genannten Dienste mithilfe von **chkconfig** aktivieren, werden die Daemons noch nicht gestartet. Verwenden Sie dazu den Befehl **/sbin/service**. Siehe Abschnitt 2.3, „Starten des **Piranha-Konfigurationstool**-Dienstes“ für ein Beispiel für die Verwendung des Befehls **/sbin/service**.

Weitere Informationen über Runlevels und die Konfiguration von Diensten mit **ntsysv** und dem **Dienst-Konfigurationstool** finden Sie im Kapitel *„Steuern des Zugriffs auf Dienste“* im *Red Hat Enterprise Linux Administrationshandbuch*.

2.2. Einrichten eines Passworts für das Piranha-Konfigurationstool

Bevor Sie das **Piranha-Konfigurationstool** zum ersten Mal auf dem primären LVS-Router verwenden, müssen Sie den Zugriff darauf mit einem Passwort einschränken. Melden Sie sich dazu als root an und führen Sie den folgenden Befehl aus:

```
/usr/sbin/piranha-passwd
```

Nachdem Sie diesen Befehl eingegeben haben, erstellen Sie das administrative Passwort, wenn Sie dazu aufgefordert werden.

Warnung

Ein sicheres Passwort sollte keine Namen, bekannte Akronyme oder Wörter aus Wörterbüchern jeglicher Sprachen enthalten. Speichern Sie das Passwort nicht unverschlüsselt auf dem System ab.

Falls das Passwort während einer aktiven Sitzung des **Piranha-Konfigurationstools** geändert wird, wird der Administrator zur Eingabe des neuen Passworts aufgefordert.

2.3. Starten des Piranha-Konfigurationstool-Dienstes

Nachdem Sie das Passwort für das **Piranha-Konfigurationstool** festgelegt haben, starten Sie den Dienst **piranha-gui** unter **/etc/rc.d/init.d/piranha-gui** bzw. starten diesen neu. Geben Sie dazu den folgenden Befehl als root ein:

```
/sbin/service piranha-gui start
```

oder

```
/sbin/service piranha-gui restart
```

Dieser Befehl startet eine private Sitzung des Apache HTTP Servers, indem der symbolische Link **/usr/sbin/piranha_gui -> /usr/sbin/httpd** aufgerufen wird. Aus Sicherheitsgründen läuft die **piranha-gui**-Version von **httpd** als piranha-Benutzer in einem separaten Prozess. Aus der Tatsache, dass **piranha-gui** den **httpd**-Dienst nutzt, ergibt sich Folgendes:

1. Der Apache HTTP Server muss auf dem System installiert sein.

2. Ein Stoppen oder Neustarten des Apache HTTP Servers mittels **service**-Befehl stoppt den **piranha-gui**-Dienst.

Warnung

Falls der Befehl **/sbin/service httpd stop** oder **/sbin/service httpd restart** auf einem LVS-Router ausgeführt wird, müssen Sie den **piranha-gui**-Dienst mithilfe des folgenden Befehls starten:

```
/sbin/service piranha-gui start
```

Lediglich der **piranha-gui**-Dienst ist notwendig, um mit der Konfiguration des Load Balancer Add-Ons zu beginnen. Falls Sie das Load Balancer Add-On jedoch von Remote aus konfigurieren, ist zudem der **sshd**-Dienst erforderlich. Sie brauchen den **pulse**-Dienst erst zu starten, wenn die Konfiguration mit dem **Piranha-Konfigurationstool** abgeschlossen ist. Siehe Abschnitt 4.8, „Starten des Load Balancer Add-Ons“ für Informationen über das Starten des **pulse**-Dienstes.

2.3.1. Konfigurieren des Piranha-Konfigurationstool-Webserver-Ports

Das **Piranha-Konfigurationstool** läuft standardmäßig auf Port 3636. Um diese Portnummer zu ändern, bearbeiten Sie die Zeile **Listen 3636** in Abschnitt 2 der **piranha-gui**-Webserver-Konfigurationsdatei **/etc/sysconfig/ha/conf/httpd.conf**.

Um das **Piranha-Konfigurationstool** zu verwenden, benötigen Sie mindestens einen textbasierten Webbrowser. Falls Sie einen Webbrowser auf dem primären LVS-Router starten, öffnen Sie die Adresse **http://*localhost*:3636**. Sie können das **Piranha-Konfigurationstool** per Webbrowser von überall erreichen, indem Sie *localhost* durch den Hostnamen oder die IP-Adresse des primären LVS-Router ersetzen.

Wenn Ihr Browser mit dem **Piranha-Konfigurationstool** verbindet, müssen Sie sich anmelden, um auf die Konfigurationsdienste zuzugreifen. Geben Sie **piranha** im Feld **Username** ein und das Passwort, das Sie mit **piranha-passwd** festgelegt haben, im Feld **Password**.

Nun, da das **Piranha-Konfigurationstool** läuft, sollten Sie in Erwägung ziehen, den Zugriff auf das Tool über das Netzwerk einzuschränken. Der nächste Abschnitt beschreibt die notwendigen Schritte dazu.

2.4. Einschränken des Zugriffs auf das Piranha-Konfigurationstool

Das **Piranha-Konfigurationstool** fragt nach einem gültigen Benutzernamen und Passwort. Da die an das **Piranha-Konfigurationstool** gesendeten Daten jedoch in Klartext übertragen werden, wird empfohlen, dass Sie den Zugriff auf vertrauenswürdige Netzwerke oder auf den lokalen Rechner einschränken.

Die einfachste Methode zur Einschränkung des Zugriffs ist die Verwendung des in Apache HTTP Server integrierten Mechanismus zur Zugriffssteuerung, bei dem Sie die Datei **/etc/sysconfig/ha/web/secure/.htaccess** bearbeiten. Nach dem Bearbeiten der Datei brauchen Sie den Dienst **piranha-gui** nicht neu zu starten, da der Server die Datei **.htaccess** jedes Mal prüft, wenn er auf das Verzeichnis zugreift.

Standardmäßig erlaubt die Zugriffssteuerung für dieses Verzeichnis jedem das Lesen der Verzeichnisinhalte. Die standardmäßigen Zugriffsrechte sehen folgendermaßen aus:

```
Order deny,allow
Allow from all
```

Um den Zugriff des **Piranha-Konfigurationstools** auf den lokalen Rechner (localhost) zu beschränken, bearbeiten Sie die Datei **.htaccess**, um Zugriff nur vom Loopback-Gerät (127.0.0.1) zu erlauben. Weitere Informationen über das Loopback-Gerät finden Sie im Kapitel *Netzwerkskripte* im *Red Hat Enterprise Linux Referenzhandbuch*.

```
Order deny,allow
Deny from all
Allow from 127.0.0.1
```

Sie können auch bestimmte Hosts oder Subnetze angeben, wie das folgende Beispiel zeigt:

```
Order deny,allow
Deny from all
Allow from 192.168.1.100
Allow from 172.16.57
```

In diesem Beispiel dürfen nur Webbrowser vom Rechner mit der IP-Adresse 192.168.1.100 sowie Rechner auf dem Netzwerk 172.16.57/24 auf das **Piranha-Konfigurationstool** zugreifen.

Warnung

Durch das Bearbeiten der Datei **.htaccess** wird der Zugriff auf die Konfigurationsseiten im Verzeichnis **/etc/sysconfig/ha/web/secure/** eingeschränkt, nicht aber auf die Anmelde- und Hilfeseiten in **/etc/sysconfig/ha/web/**. Um den Zugriff auf dieses Verzeichnis einzuschränken, erstellen Sie eine **.htaccess**-Datei im Verzeichnis **/etc/sysconfig/ha/web/** mit den Zeilen **order**, **allow** und **deny** genau wie in **/etc/sysconfig/ha/web/secure/.htaccess**.

2.5. Aktivieren der Paketweiterleitung

Damit der LVS-Router Netzwerkpakete ordnungsgemäß an die realen Server weiterleitet, muss auf jedem LVS-Router die IP-Weiterleitung im Kernel aktiviert sein. Melden Sie sich als root an und ändern Sie die Zeile **net.ipv4.ip_forward = 0** in **/etc/sysctl.conf** folgendermaßen:

```
net.ipv4.ip_forward = 1
```

Diese Änderung wird nach dem nächsten Neustart des Systems wirksam.

Um zu überprüfen, ob die IP-Weiterleitung aktiviert ist, führen Sie den folgenden Befehl als root aus:

```
/sbin/sysctl net.ipv4.ip_forward
```

Falls der obige Befehl **1** ausgibt, ist die IP-Weiterleitung aktiviert. Falls er **0** ausgibt, können Sie die IP-Weiterleitung manuell mithilfe des folgenden Befehls aktivieren:

```
/sbin/sysctl -w net.ipv4.ip_forward=1
```

2.6. Konfigurieren von Diensten auf den realen Servern

Falls es sich bei den realen Servern um Red Hat Enterprise Linux Systeme handelt, stellen Sie die entsprechenden Server-Daemons zum Start beim Systemboot ein. Zu diesen Daemons können unter anderem **`httpd`** für Webdienste oder **`xinetd`** für FTP- oder Telnet-Dienste gehören.

Unter Umständen kann es hilfreich sein, auf die realen Server von Remote aus zugreifen zu können, weshalb der **`sshd`**-Daemon ebenfalls installiert und aktiviert sein sollte.

Kapitel 3. Einrichten des Load Balancer Add-Ons

Das Load Balancer Add-On besteht aus zwei grundlegenden Gruppen: die LVS-Router und die realen Server. Um die Entstehung eines Single Point of Failure zu vermeiden, sollte jede Gruppe mindestens zwei Systeme umfassen.

Die LVS-Routergruppe sollte aus zwei identischen oder sehr ähnlichen Systemen bestehen, auf denen Red Hat Enterprise Linux läuft. Eines der Systeme agiert als aktiver LVS-Router, während das andere im Bereitschaftsmodus bleibt, weshalb beide über etwa dieselben Spezifikationen verfügen sollten.

Bevor Sie die Hardware für die Gruppe der realen Server auswählen und konfigurieren, entscheiden Sie sich für eine der drei Load Balancer Add-On Topologien.

3.1. Das NAT Load Balancer Add-On Netzwerk

Die NAT-Topologie ermöglicht einen größeren Spielraum bei der Verwendung vorhandener Hardware, ist jedoch eingeschränkt in dessen Fähigkeit zur Handhabung großer Arbeitslasten, da alle Pakete auf ihrem Weg in den Pool oder aus dem Pool den Load Balancer Add-On Router durchqueren müssen.

Netzwerkaufbau

Die Topologie für das Load Balancer Add-On mit NAT-Routing ist im Hinblick auf den Netzwerkaufbau am einfachsten zu konfigurieren, da nur ein Zugangspunkt zum öffentlichen Netzwerk benötigt wird. Die realen Server senden alle Anfragen zurück über den LVS-Router, befinden sich also in ihrem eigenen, privaten Netzwerk.

Hardware

Die NAT-Topologie ist im Hinblick auf die Hardware am flexibelsten, da es sich bei den realen Servern nicht um Linux-Rechner handeln muss. In einer NAT-Topologie benötigt jeder reale Server nur eine NIC, da er nur mit dem LVS-Router kommuniziert. Die LVS-Router dagegen benötigen jeweils zwei NICs, um den Datenverkehr zwischen den Netzwerken hin- und herzuleiten. Da diese Topologie einen Engpass am LVS-Router verursacht, können Gigabit-Ethernet-NICs auf jedem LVS-Router eingesetzt werden, um die Bandbreite der LVS-Router zu erhöhen. Falls Gigabit-Ethernet auf den LVS-Routern eingesetzt wird, muss jeder Switch, der die realen Server mit den LVS-Routern verbindet, mindestens über zwei Gigabit-Ethernet-Ports verfügen, um die Last effizient zu handhaben.

Software

Da die NAT-Topologie die Verwendung von **`iptables`** für einige Konfigurationen erfordert, ist unter Umständen einiges an Software-Konfiguration außerhalb des **Piranha-Konfigurationstools** nötig. Insbesondere FTP-Dienste und die Verwendung von Firewall-Markierungen erfordern eine zusätzliche, manuelle Konfiguration der LVS-Router, um Anfragen ordnungsgemäß weiterzuleiten.

3.1.1. Konfigurieren von Netzwerkschnittstellen für Load Balancer Add-On mit NAT

Um das Load Balancer Add-On mit NAT einzurichten, müssen Sie zunächst die Netzwerkschnittstellen für das öffentliche Netzwerk und das private Netzwerk auf den LVS-Routern einrichten. In diesem Beispiel befinden sich die öffentlichen Schnittstellen des LVS-Routers (**`eth0`**) auf dem Netzwerk 192.168.26/24 (dies ist keine routbare IP, gehen Sie jedoch von einer Firewall vor

dem LVS-Router aus). Die privaten Schnittstellen, die mit den realen Servern verbinden (**eth1**), befinden sich auf dem Netzwerk 10.11.12/24.

Wichtig

Beachten Sie, dass die folgenden Dateien zum **network**-Dienst gehören und dass das Load Balancer Add-On nicht kompatibel ist mit dem **NetworkManager**-Dienst.

Auf dem aktiven oder *primären* LVS-Routerknoten kann das Netzwerkskript der öffentlichen Schnittstelle, **/etc/sysconfig/network-scripts/ifcfg-eth0**, etwa wie folgt aussehen:

```
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.26.9
NETMASK=255.255.255.0
GATEWAY=192.168.26.254
```

Das Netzwerkskript **/etc/sysconfig/network-scripts/ifcfg-eth1** für die private NAT-Schnittstelle auf dem LVS-Router könnte etwa wie folgt aussehen:

```
DEVICE=eth1
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.11.12.9
NETMASK=255.255.255.0
```

In diesem Beispiel lautet die VIP für die öffentliche Schnittstelle des LVS-Routers 192.168.26.10 und die VIP für die NAT- oder private Schnittstelle lautet 10.11.12.10. Somit ist es unumgänglich, dass die realen Server ihre Anfragen zurück an die VIP für die NAT-Schnittstelle leiten.

Wichtig

Die beispielhaften Konfigurationseinstellungen der Ethernet-Schnittstelle in diesem Abschnitt sind für die realen IP-Adressen eines LVS-Routers und *nicht* für die Floating-IP-Adressen. Um die öffentlichen und privaten Floating-IP-Adressen zu konfigurieren, sollte der Administrator das **Piranha-Konfigurationstool** verwenden, wie in Abschnitt 4.4, „**GLOBAL SETTINGS**“ und Abschnitt 4.6.1, „Der Unterabschnitt **VIRTUAL SERVER**“ beschrieben.

Konfigurieren Sie nach den Netzwerkschnittstellen des primären LVS-Routerknotens nun die realen Netzwerkschnittstellen des Backup-LVS-Routers. Vergewissern Sie sich dabei, dass keine der IP-Adressen mit anderen IP-Adressen auf dem Netzwerk kollidiert.

Wichtig

Stellen Sie sicher, dass jede Schnittstelle auf dem Backup-Knoten demselben Netzwerk angehört wie die Schnittstelle auf dem primären Knoten. Falls beispielsweise eth0 auf dem primären Knoten mit dem öffentlichen Netzwerk verbindet, muss eth0 auf dem Backup-Knoten ebenfalls mit dem öffentlichen Netzwerk verbinden.

3.1.2. Routing auf den realen Servern

Das wichtigste bei der Konfiguration der Netzwerkschnittstellen für die realen Server in einer NAT-Topologie ist die Einstellung des Gateways für die NAT-Floating-IP-Adresse des LVS-Routers. In diesem Beispiel lautet diese Adresse 10.11.12.10.

Anmerkung

Sobald die Netzwerkschnittstellen auf den realen Servern aktiv sind, werden sich die Rechner nicht mehr auf andere Weise mit dem öffentlichen Netzwerk verbinden oder es anpingen können. Dies ist normal. Sie können jedoch die reale IP-Adresse für die private Schnittstelle des LVS-Routers anpingen, in diesem Fall 10.11.12.9.

Die Datei **/etc/sysconfig/network-scripts/ifcfg-eth0** des realen Servers sollte also etwa wie folgt aussehen:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.11.12.1
NETMASK=255.255.255.0
GATEWAY=10.11.12.10
```

Warnung

Falls ein realer Server mehr als eine Netzwerkschnittstelle mit der Zeile **GATEWAY=** konfiguriert hat, wird die Netzwerkschnittstelle, die als erstes aktiv wird, das Gateway erhalten. Falls sowohl **eth0** als auch **eth1** konfiguriert sind und **eth1** für das Load Balancer Add-On verwendet wird, könnten Anfragen von den realen Servern infolgedessen unter Umständen nicht korrekt geroutet werden.

Es empfiehlt sich, überflüssige Netzwerkschnittstellen zu deaktivieren, indem Sie in deren Netzwerkskript im Verzeichnis **/etc/sysconfig/network-scripts/** die Option **ONBOOT=no** angeben. Andernfalls sollten Sie sicherstellen, dass das Gateway korrekt konfiguriert ist in derjenigen Schnittstelle, die zuerst aktiviert wird.

3.1.3. Aktivieren von NAT-Routing auf den LVS-Routern

In einer einfachen Load Balancer Add-On Konfiguration mit NAT, in der jeder geclusterte Dienst nur einen Port verwendet, wie z. B. HTTP auf Port 80, muss der Administrator lediglich die Paketweiterleitung auf den LVS-Routern aktivieren, damit die Anfragen ordnungsgemäß zwischen den realen Servern und dem öffentlichen Netzwerk geroutet werden. In Abschnitt 2.5, „Aktivieren der Paketweiterleitung“ finden Sie Anweisungen zum Aktivieren der Paketweiterleitung. Wenn die geclusterten Dienste jedoch erfordern, dass mehr als ein Port während derselben Benutzersitzung an denselben realen Server geleitet wird, sind mehr Konfigurationsschritte erforderlich. Weitere Informationen über das Erstellen von Multi-Port-Diensten mithilfe von Firewall-Markierungen finden Sie in Abschnitt 3.4, „Multi-Port-Dienste und Load Balancer Add-On“.

Nachdem die Weiterleitung auf den LVS-Routern aktiviert wurde und die realen Server eingerichtet sind und die geclusterten Dienste ausführen, verwenden Sie das **Piranha-Konfigurationstool** zur Konfiguration des Load Balancer Add-Ons wie in Kapitel 4, *Konfigurieren des Load Balancer Add-Ons mit dem **Piranha-Konfigurationstool*** beschrieben.

Warnung

Konfigurieren Sie die Floating-IP für **eth0:1** oder **eth1:1** nicht durch manuelles Bearbeiten von Netzwerkskripten oder mit einem Netzwerkkonfigurationstool. Verwenden Sie stattdessen das **Piranha-Konfigurationstool**, wie in Abschnitt 4.4, „**GLOBAL SETTINGS**“ und Abschnitt 4.6.1, „Der Unterabschnitt **VIRTUAL SERVER**“ beschrieben.

Wenn Sie fertig sind, starten Sie den **pulse**-Dienst wie in Abschnitt 4.8, „Starten des Load Balancer Add-Ons“ beschrieben. Sobald **pulse** läuft, beginnt der aktive LVS-Router mit der Weiterleitung von Anfragen an den Pool der realen Server.

3.2. Load Balancer Add-On mit direktem Routing

Wie in Abschnitt 1.4.2, „Direktes Routing“ erwähnt, ermöglicht das direkte Routing den realen Servern, Pakete zu verarbeiten und direkt an den anfragenden Benutzer zu leiten, statt ausgehende Pakete durch den LVS-Router leiten zu müssen. Direktes Routing erfordert, dass die realen Server physisch mit einem Netzwerksegment mit dem LVS-Router verbunden sind und ausgehende Pakete verarbeitet und weitergeleitet werden können.

Netzwerkaufbau

In einer Load Balancer Add-On Konfiguration mit direktem Routing muss der LVS-Router eingehende Anfragen empfangen und diese an die richtigen realen Server zur Verarbeitung weiterleiten. Die realen Server müssen ihre Antwort dann *direkt* an den Client senden. Falls sich der Client beispielsweise im Internet befindet und das Paket durch den LVS-Router an einen realen Server sendet, dann muss der reale Server dazu in der Lage sein, über das Internet direkt an den Client zu senden. Sie erreichen dies, indem Sie ein Gateway für den realen Server konfigurieren, das die Pakete an das Internet weiterleitet. Jeder reale Server im Server-Pool kann über ein eigenes Gateway verfügen (und jedes Gateway über eine eigene Verbindung zum Internet) für maximalen Durchsatz und optimale Skalierbarkeit. In typischen Load Balancer Add-On Konfigurationen können die realen Server jedoch über ein einziges Gateway und somit eine Netzwerkverbindung kommunizieren.

Wichtig

Es wird nicht empfohlen, den LVS-Router als Gateway für die realen Server zu verwenden, da dies die Konfiguration des LVS-Routers unnötig verkompliziert und diesem darüber hinaus weitere Netzwerklast aufbürdet, wodurch wie beim NAT-Routing wieder ein Engpass entsteht.

Hardware

Die Hardware-Anforderungen eines Load Balancer Add-On Systems mit direktem Routing ähneln denen anderer Load Balancer Add-On Topologien. Während auf dem LVS-Router Red Hat Enterprise Linux laufen muss, um die eingehenden Anfragen zu verarbeiten und die Lastverteilung für die realen Server durchzuführen, muss es sich bei den realen Servern nicht um Linux-Rechner handeln, um ordnungsgemäß zu funktionieren. Die LVS-Router

benötigen jeweils eine oder zwei NICs (abhängig davon, ob es einen Backup-Router gibt). Sie können zwei NICs verwenden, um die Konfiguration zu vereinfachen und den Datenverkehr abzugrenzen – eingehende Anfragen werden von einer NIC gehandhabt und geroutete Pakete zu den realen Servern auf der anderen.

Da die realen Server den LVS-Router umgehen und ausgehende Pakete direkt an einen Client senden, ist ein Gateway zum Internet erforderlich. Für ein Höchstmaß an Leistung und Verfügbarkeit kann jeder reale Server mit einem eigenen Gateway verbunden werden, der über eine eigene Verbindung zu dem Netzwerk verfügt, auf dem sich der Client befindet (z. B. das Internet oder ein Intranet).

Software

Es gibt einige notwendige Konfigurationsschritte außerhalb des **Piranha-Konfigurationstools**, insbesondere für Administratoren, die beim Einsatz des Load Balancer Add-Ons mit direktem Routing auf ARP-Probleme stoßen. Siehe Abschnitt 3.2.1, „Direktes Routing und **arptables_jf**“ oder Abschnitt 3.2.2, „Direktes Routing und **iptables**“ für weitere Informationen.

3.2.1. Direktes Routing und `arptables_jf`

Um direktes Routing mithilfe von **arptables_jf** zu konfigurieren, muss auf jedem realen Server eine virtuelle IP-Adresse konfiguriert sein, damit diese Pakete direkt routen können. ARP-Anfragen für die VIP werden von den realen Servern völlig ignoriert und jegliche andere ARP-Pakete, die gegebenenfalls die VIPs enthalten, werden geändert, um die IP des realen Servers zu enthalten anstelle der VIPs.

Mithilfe von **arptables_jf** können Applikationen mit jeder einzelnen VIP oder jedem einzelnen Port verbinden, die vom realen Server bereitgestellt werden. Beispielsweise ermöglicht **arptables_jf**, dass mehrere laufende Instanzen von Apache HTTP Server explizit mit anderen VIPs auf dem System verknüpft werden. Es gibt zudem deutliche Leistungsvorteile bei der Verwendung von **arptables_jf** im Vergleich zur **iptables**-Option.

Allerdings ist es bei Verwendung von **arptables_jf** nicht möglich, mithilfe von standardmäßigen Red Hat Enterprise Linux Systemkonfigurationstools die VIPs zum Start beim Bootzeitpunkt zu konfigurieren.

Führen Sie die folgenden Schritte aus, um jeden realen Server zu konfigurieren, um ARP-Anfragen für jede virtuelle IP-Adresse zu ignorieren:

1. Erstellen Sie die ARP-Tabelleneinträge für jede virtuelle IP-Adresse auf jedem realen Server (die real_ip ist die IP, die der Router zur Kommunikation mit dem realen Server verwendet, oft ist diese IP **eth0** zugewiesen):

   ```
   arptables -A IN -d <virtual_ip> -j DROP
   arptables -A OUT -s <virtual_ip> -j mangle --mangle-ip-s <real_ip>
   ```

   Dies führt dazu, dass die realen Server alle ARP-Anfragen für die virtuellen IP-Adressen ignorieren, und ändert jegliche ausgehende ARP-Antworten, die andernfalls die virtuelle IP enthalten, auf die reale IP des Servers. Der einzige Knoten, der auf ARP-Anfragen für die VIPs reagieren sollte, ist der derzeit aktive LVS-Knoten.

2. Sobald dies auf jedem realen Server fertiggestellt wurde, speichern Sie die ARP-Tabelleneinträge, indem Sie die folgenden Befehle auf jedem realen Server ausführen:

   service arptables_jf save

```
chkconfig --level 2345 arptables_jf on
```

Der **chkconfig**-Befehl führt dazu, dass das System die arptables-Konfiguration beim Systemstart – noch vor dem Start des Netzwerks – neu lädt.

3. Konfigurieren Sie die virtuelle IP-Adresse auf allen realen Servern mithilfe von **ifconfig**, um ein IP-Alias zu erstellen. Zum Beispiel:

```
# ifconfig eth0:1 192.168.76.24 netmask 255.255.252.0 broadcast
192.168.79.255 up
```

Sie können auch das **iproute2**-Dienstprogramm **ip** verwenden, zum Beispiel:

```
# ip addr add 192.168.76.24 dev eth0
```

Wie bereits erwähnt, können die virtuellen IP-Adressen nicht mithilfe der Red Hat Systemkonfigurationstools zum Starten beim Systemstart konfiguriert werden. Sie können dieses Problem jedoch umgehen, indem Sie diese Befehle in die Datei **/etc/rc.d/rc.local** einfügen.

4. Konfigurieren Sie Piranha für das direkte Routing. Siehe Kapitel 4, *Konfigurieren des Load Balancer Add-Ons mit dem **Piranha-Konfigurationstool*** für weitere Informationen.

3.2.2. Direktes Routing und `iptables`

Sie können das ARP-Problem mit der direkten Routing-Methode auch umgehen, indem Sie **iptables**-Firewall-Regeln erstellen. Um das direkte Routing mit **iptables** zu konfigurieren, müssen Sie Regeln hinzufügen, die einen transparenten Proxy erstellen, sodass ein realer Server die an die VIP-Adresse gesendeten Pakete verarbeitet, obwohl die VIP-Adresse nicht auf dem System existiert.

Die **iptables**-Methode ist einfacher zu konfigurieren als die **arptables_jf**-Methode. Zudem umgeht diese Methode das LVS-ARP-Problem gänzlich, da die virtuellen IP-Adressen nur auf dem aktiven LVS-Router existieren.

Allerdings bringt die **iptables**-Methode im Vergleich zu **arptables_jf** Leistungsnachteile mit sich, da die Weiterleitung und Maskierung jedes Pakets einen Overhead verursacht.

Zudem können Sie mit der **iptables**-Methode keine Ports wiederverwenden. Beispielsweise ist es nicht möglich, zwei separate Apache HTTP Server-Dienste mit Port 80 zu verknüpfen, da beide mit ***INADDR_ANY*** verbunden werden müssen anstelle der virtuellen IP-Adressen.

Führen Sie die folgenden Schritte aus, um direktes Routing mit der **iptables**-Methode zu konfigurieren:

1. Führen Sie auf jedem realen Server den folgenden Befehl für jede Kombination aus VIP, Port und Protokoll (TCP oder UDP) aus, die von den realen Servern bedient werden soll:

```
iptables -t nat -A PREROUTING -p <tcp|udp> -d <vip> --dport <port> -j
REDIRECT
```

Dieser Befehl veranlasst die realen Server dazu, Pakete zu verarbeiten, die auf die angegebenen VIPs und Ports abzielen.

2. Speichern Sie die Konfiguration auf jedem realen Server:

```
# service iptables save
# chkconfig --level 2345 iptables on
```

Die obigen Befehle veranlassen das System dazu, die **iptables**-Konfiguration beim Systemstart – noch vor dem Start des Netzwerks – neu zu laden.

3.3. Zusammenstellen der Konfiguration

Nachdem Sie sich entschieden haben, welche der oben genannten Routing-Methoden Sie verwenden möchten, sollten Sie die Hardware auf dem Netzwerk zusammenschließen.

Wichtig

Die Adaptergeräte auf den LVS-Routern müssen konfiguriert werden, um auf dieselben Netzwerke zuzugreifen. Falls **eth0** beispielsweise mit dem öffentlichen Netzwerk verbindet und **eth1** mit dem privaten Netzwerk, dann müssen auf dem Backup-LVS-Router dieselben Geräte mit denselben Netzwerken verbinden.

Das Gateway, das in der ersten Schnittstelle angegeben ist, die beim Systemstart aktiv wird, wird zur Routing-Tabelle hinzugefügt. Nachfolgende Gateways, die in anderen Schnittstellen aufgeführt sind, werden ignoriert. Dies ist insbesondere wichtig bei der Konfiguration der realen Server.

Nachdem die Hardware physisch miteinander verbunden ist, konfigurieren Sie die Netzwerkschnittstellen auf den primären und den Backup-LVS-Routersn. Sie erreichen dies entweder mit einer grafischen Applikation wie **system-config-network** oder durch manuelles Bearbeiten der Netzwerkskripte. Weitere Informationen über das Hinzufügen von Geräten mittels **system-config-network** finden Sie im Kapitel namens *Netzwerkkonfiguration* im *Red Hat Enterprise Linux Bereitstellungshandbuch*. Im Verlauf dieses Kapitels werden die beispielhaften Änderungen an Netzwerkschnittstellen entweder manuell oder mithilfe des **Piranha-Konfigurationstools** vorgenommen.

3.3.1. Allgemeine Tipps für ein Netzwerk mit Load Balancer Add-On

Konfigurieren Sie die realen IP-Adressen für die öffentlichen und privaten Netzwerke auf den LVS-Routern, bevor Sie mit der Konfiguration des Load Balancer Add-Ons mithilfe des **Piranha-Konfigurationstool** beginnen. Die Abschnitte über die jeweilige Topologie zeigen beispielhafte Netzwerkadressen, für Ihre Konfiguration benötigen Sie jedoch die tatsächlichen Netzwerkadressen. Nachfolgend sehen Sie einige hilfreiche Befehle zur Aktivierung von Netzwerkschnittstellen und zur Statusprüfung.

Aktivieren von Netzwerkschnittstellen

Um eine reale Netzwerkschnittstelle zu aktivieren, führen Sie den folgenden Befehl als root-Benutzer aus und ersetzen Sie dabei *N* durch die Nummer Ihrer Schnittstelle (**eth0** und **eth1**).

```
/sbin/ifup ethN
```

Warnung

Verwenden Sie die **ifup**-Skripte *nicht* zur Aktivierung von Floating-IP-Adressen, die Sie ggf. mit dem **Piranha-Konfigurationstool** erstellen (**eth0:1** oder **eth1:1**). Verwenden Sie stattdessen den **service**-Befehl zum Starten von **pulse** (siehe Abschnitt 4.8, „Starten des Load Balancer Add-Ons“ für Einzelheiten).

Deaktivieren von Netzwerkschnittstellen

Um eine reale Netzwerkschnittstelle zu deaktivieren, führen Sie den folgenden Befehl als root-Benutzer aus und ersetzen Sie dabei *N* durch die Nummer Ihrer Schnittstelle (**eth0** und **eth1**).

```
/sbin/ifdown ethN
```

Überprüfen des Status der Netzwerkschnittstellen

Falls Sie prüfen möchten, welche Netzwerkschnittstellen derzeit aktiv sind, geben Sie Folgendes ein:

```
/sbin/ifconfig
```

Um die Routing-Tabelle für einen Rechner einzusehen, führen Sie den folgenden Befehl aus:

```
/sbin/route
```

3.3.1.1. Suche und Bereinigung von Fehlern mit virtuellen IP-Adressen

In einigen Situationen kann es vorkommen, dass ein Administrator auf Probleme trifft beim automatischen Wechsel (Failover) von einem aktiven LVS-Host auf den Standby-Host. Unter Umständen werden beim Failover nicht alle virtuellen IP-Adressen auf dem Standby-Host aktiviert. Dieses Problem kann auch auftreten, wenn der Standby-Host gestoppt und der primäre Host aktiviert wird. Nur wenn der **pulse**-Dienst manuell neu gestartet wird, werden alle virtuellen IP-Adressen aktiviert.

Um dieses Problem vorübergehend zu lösen, führen Sie den folgenden Befehl an der root-Shell-Eingabeaufforderung aus:

```
echo 1 > /proc/sys/net/ipv4/conf/all/promote_secondaries
```

Beachten Sie, dass dies nur eine *vorübergehende* Lösung des Problems ist und dass der Befehl einen Systemneustart nicht überdauert.

Um dieses Problem permanent zu lösen, öffnen Sie die Datei **/etc/sysctl.conf** und fügen Sie die folgende Zeile hinzu:

```
net.ipv4.conf.all.promote_secondaries = 1
```

3.4. Multi-Port-Dienste und Load Balancer Add-On

In jeder Topologie erfordern LVS-Router weitere Konfigurationsschritte, wenn Multi-Port-Dienste für das Load Balancer Add-On erstellt werden sollen. Multi-Port-Dienste können künstlich mithilfe von

Firewall-Markierungen erzeugt werden, indem verschiedene, aber verwandte Protokolle wie z. B. HTTP (Port 80) und HTTPS (Port 443) gebündelt werden, oder wenn das Load Balancer Add-On mit realen Multi-Port-Protokollen wie z. B. FTP verwendet wird. In beiden Fällen verwendet der LVS-Router Firewall-Markierungen, um zu kennzeichnen, dass Pakete zwar für verschiedene Ports vorgesehen sind, aufgrund derselben Firewall-Markierung jedoch identisch gehandhabt werden sollen. In Kombination mit Persistenz gewährleisten Firewall-Markierungen, dass Verbindungen vom Client-Rechner zu demselben Host geleitet werden, sofern die Verbindung innerhalb der vom Persistenzparameter festgelegten Zeitspanne erfolgt. Weitere Informationen über das Zuweisen von Persistenz zu einem virtuellen Server finden Sie in Abschnitt 4.6.1, „Der Unterabschnitt **VIRTUAL SERVER**“.

Leider kann der Mechanismus zur Lastverteilung auf den realen Servern – IPVS – zwar die einem Paket zugewiesenen Firewall-Markierungen erkennen, kann jedoch selbst keine Firewall-Markierungen zuweisen. Die Aufgabe des *Zuweisens* von Firewall-Markierungen muss vom Netzwerkpaketfilter **iptables** außerhalb des **Piranha-Konfigurationstools** erfolgen.

3.4.1. Zuweisen von Firewall-Markierungen

Der Administrator muss zum Zuweisen von Firewall-Markierungen zu einem Paket, das für einen bestimmten Port vorgesehen ist, **iptables** verwenden.

Dieser Abschnitt veranschaulicht, wie beispielsweise HTTP und HTTPS gebündelt werden; auch FTP ist ein häufig geclustertes Multi-Port-Protokoll. Falls ein Load Balancer Add-On für FTP-Dienste verwendet wird, finden Sie in Abschnitt 3.5, „Konfigurieren von FTP“ die Konfigurationsdetails.

Die wichtigste Regel bei der Verwendung von Firewall-Markierungen ist die, dass es für jedes Protokoll, das im **Piranha-Konfigurationstool** eine Firewall-Markierung verwendet, eine entsprechende **iptables**-Regel geben muss, um den Netzwerkpaketen die Markierung zuzuweisen.

Bevor Sie Netzwerkpaketfilterregeln erstellen, vergewissern Sie sich, dass nicht bereits Regeln vorhanden sind. Öffnen Sie dazu eine Shell-Eingabeaufforderung, melden Sie sich als root-Benutzer an und geben Sie Folgendes ein:

```
/sbin/service iptables status
```

Falls **iptables** nicht ausgeführt wird, kehrt die Eingabeaufforderung sofort zurück.

Falls **iptables** aktiv ist, werden eine Reihe von Regeln angezeigt. Falls Regeln vorhanden sind, geben Sie den folgenden Befehl ein:

```
/sbin/service iptables stop
```

Falls die vorhandenen Regeln wichtig sind, prüfen Sie den Inhalt der Datei **/etc/sysconfig/iptables** und kopieren Sie jegliche Regeln, die Sie behalten möchten, an einen sicheren Speicherort, bevor Sie fortfahren.

Nachfolgend sehen Sie die Regeln, die dem eingehenden Datenverkehr für die Floating-IP-Adresse *n.n.n.n* dieselbe Firewall-Markierung 80 zuweist, auf Ports 80 und 443.

```
/sbin/iptables -t mangle -A PREROUTING -p tcp -d n.n.n.n/32 -m multiport
--dports 80,443 -j MARK --set-mark 80
```

Siehe Abschnitt 4.6.1, „Der Unterabschnitt **VIRTUAL SERVER**“ für Anweisungen zur Zuweisung der VIP zur öffentlichen Netzwerkschnittstelle. Beachten Sie, dass Sie sich als root-Benutzer anmelden und das **iptables**-Modul laden müssen, bevor Sie zum ersten Mal Regeln erstellen.

In den obigen **`iptables`**-Befehlen sollte *n.n.n.n* durch die Floating-IP für Ihre virtuellen HTTP- und HTTPS-Server ersetzt werden. Diese Befehle weisen effektiv jeglichem Datenverkehr, der an die VIP auf den entsprechenden Ports gerichtet ist, die Firewall-Markierung 80 zu, die anschließend von IPVS erkannt und entsprechend weitergeleitet wird.

Warnung

Die obigen Befehle werden sofort wirksam, überdauern jedoch keinen Neustart des Systems. Um sicherzugehen, dass die Einstellungen des Netzwerkpaketfilters nach einem Neustart wiederhergestellt werden, folgen Sie den Anweisungen in Abschnitt 3.6, „Speichern der Netzwerkpaketfiltereinstellungen“

3.5. Konfigurieren von FTP

Das File Transport Protocol (FTP) ist ein altes und komplexes Multi-Port-Protokoll, das eine Reihe besonderer Herausforderungen an eine Load Balancer Add-On Umgebung stellt. Um diese Herausforderungen zu verstehen, müssen Sie sich zunächst die wesentlichen Merkmale der FTP-Funktionsweise vergegenwärtigen.

3.5.1. Funktionsweise von FTP

Bei den meisten anderen Client-Server-Beziehungen öffnet der Client-Rechner eine Verbindung zum Server auf einem bestimmten Port und der Server antwortet dem Client auf demselben Port. Wenn sich ein FTP-Client mit einem FTP-Server verbindet, öffnet er eine Verbindung auf dem FTP-Steuerungsport 21. Anschließend teilt der *Client* dem FTP-*Server* mit, ob eine *aktive* oder *passive* Verbindung hergestellt werden soll. Die Art der vom Client gewählten Verbindung bestimmt, wie der Server antwortet und auf welchen Ports die Datenübertragung stattfindet.

Es gibt zwei Arten von Datenverbindungen:

Aktive Verbindungen

Wenn eine aktive Verbindung hergestellt wird, öffnet der *Server* eine Datenverbindung zum Client von Port 20 zu einem Port im hohen Bereich auf dem Client-Rechner. Sämtliche Daten vom Server werden über diese Verbindung übertragen.

Passive Verbindungen

Wenn eine passive Verbindung hergestellt wird, fordert der *Client* den FTP-Server zur Herstellung eines passiven Verbindungsports auf, der auf einem beliebigen Port über 10.000 liegen kann. Der Server verbindet für diese Sitzung mit diesem hohen Port und teilt dem Client die Portnummer mit. Der Client öffnet daraufhin den neu verbundenen Port für die Datenverbindung. Jede Datenanforderung durch den Client resultiert in einer separaten Datenverbindung. Moderne FTP-Clients versuchen, bei der Anforderung von Daten von Servern eine passive Verbindung zu erstellen.

Anmerkung

Der *Client* bestimmt die Art der Verbindung, nicht der Server. Infolgedessen müssen Sie für eine effektive Clusterung von FTP Ihre LVS-Router zur Handhabung von sowohl aktiven als auch passiven Verbindungen konfigurieren.

Aufgrund dieser Client-Server-Beziehung von FTP werden möglicherweise eine große Anzahl an Ports geöffnet, von denen das **Piranha-Konfigurationstool** und IPVS keine Kenntnis haben.

3.5.2. Auswirkungen auf das Load Balancer Add-On Routing

Die IPVS-Paketweiterleitung erlaubt Verbindungen zum bzw. aus dem Cluster basierend darauf, ob es die Portnummer oder die Firewall-Markierung erkennt. Falls ein Client von außerhalb des Clusters versucht, einen Port zu öffnen, für dessen Handhabung IPVS nicht konfiguriert ist, wird die Verbindung abgelehnt. Umgekehrt wird die Verbindung ebenfalls abgelehnt, wenn ein realer Server versucht, eine Verbindung zum Internet auf einem Port herzustellen, der IPVS unbekannt ist. Dies bedeutet, dass *alle* Verbindungen von FTP-Clients im Internet dieselbe Firewall-Markierung tragen *müssen* und dass alle Verbindungen vom FTP-Server mithilfe von Netzwerkpaketfilterregeln ordnungsgemäß an das Internet weitergeleitet werden *müssen*.

Anmerkung

Um passive FTP-Verbindungen zu ermöglichen, vergewissern Sie sich, dass das Kernelmodul **ip_vs_ftp** geladen ist. Führen Sie dazu den Befehl **modprobe ip_vs_ftp** als root-Benutzer an einer Shell-Eingabeaufforderung aus.

3.5.3. Erstellen von Netzwerkpaketfilterregeln

Bevor Sie **iptables**-Regeln für den FTP-Dienst erstellen, werfen Sie einen Blick auf die Informationen in Abschnitt 3.4.1, „Zuweisen von Firewall-Markierungen“ über Multi-Port-Dienste und Techniken zur Prüfung der vorhandenen Netzwerkpaketfilterregeln.

Nachfolgend sehen Sie die Regeln, die dem FTP-Datenverkehr dieselbe Firewall-Markierung (21) zuweist. Damit diese Regeln ordnungsgemäß funktionieren, müssen Sie zudem auf dem Unterreiter **VIRTUAL SERVER** des **Piranha-Konfigurationstools** einen virtuellen Server für Port 21 mit dem Wert **21** im Feld **Firewall Mark** konfigurieren. Siehe Abschnitt 4.6.1, „Der Unterabschnitt **VIRTUAL SERVER**“ für Einzelheiten.

3.5.3.1. Regeln für aktive Verbindungen

Die Regeln für aktive Verbindungen weisen den Kernel an, Verbindungen zur *internen* Floating-IP-Adresse auf Port 20 (dem FTP-Datenport) zu akzeptieren und weiterzuleiten.

Der folgende **iptables**-Befehl erlaubt es dem LVS-Router, ausgehende Verbindungen von den realen Servern zu akzeptieren, die IPVS unbekannt sind:

```
/sbin/iptables -t nat -A POSTROUTING -p tcp -s n.n.n.0/24 --sport 20 -j MASQUERADE
```

Ersetzen Sie *n.n.n* im obigen **iptables**-Befehl durch die ersten drei Werte der Floating-IP für die interne Netzwerkschnittstelle der NAT-Schnittstelle, wie auf dem Reiter **GLOBAL SETTINGS** im **Piranha-Konfigurationstool** definiert.

3.5.3.2. Regeln für passive Verbindungen

Die Regeln für passive Verbindungen weisen die richtige Firewall-Markierung für Verbindungen zu, die vom Internet an die Floating-IP für den Dienst auf einem großen Bereich von Ports (10.000 bis 20.000) gehen.

Warnung

Falls Sie den Portbereich für passive Verbindungen einschränken, müssen Sie auch den VSFTP-Server zur Verwendung desselben Portbereichs konfigurieren. Sie erreichen dies, indem Sie die folgenden Zeilen zur **/etc/vsftpd.conf** hinzufügen:

```
pasv_min_port=10000

pasv_max_port=20000
```

Die Option **pasv_address** zum Überschreiben der realen FTP-Serveradresse sollte nicht verwendet werden, da sie von LVS auf die virtuelle IP-Adresse aktualisiert wird.

Weitere Informationen über die Konfiguration von anderen FTP-Servern finden Sie in der entsprechenden Dokumentation.

Dieser Bereich sollte für die meisten Anwendungsfälle groß genug sein; bei Bedarf können Sie ihn jedoch vergrößern, um alle verfügbaren nicht sicheren Ports einzubeziehen, indem Sie **10000:20000** in den nachfolgenden Befehlen auf **1024:65535** ändern.

Diese folgenden **iptables**-Befehle weisen effektiv jeglichem Datenverkehr, der an die Floating-IP auf den entsprechenden Ports gerichtet ist, die Firewall-Markierung 21 zu, die anschließend von IPVS erkannt und entsprechend weitergeleitet wird:

```
/sbin/iptables -t mangle -A PREROUTING -p tcp -d n.n.n.n/32 --dport 21 -j
MARK --set-mark 21
```

```
/sbin/iptables -t mangle -A PREROUTING -p tcp -d n.n.n.n/32 --dport
10000:20000 -j MARK --set-mark 21
```

Ersetzen Sie in den **iptables**-Befehlen *n.n.n.n* durch die Floating-IP für den virtuellen FTP-Server, der in dem Unterbereich **VIRTUAL SERVER** im **Piranha-Konfigurationstool** definiert ist.

Warnung

Die obigen Befehle werden sofort wirksam, überdauern jedoch keinen Neustart des Systems. Um sicherzugehen, dass die Einstellungen des Netzwerkpaketfilters nach einem Neustart wiederhergestellt werden, folgen Sie den Anweisungen in Abschnitt 3.6, „Speichern der Netzwerkpaketfiltereinstellungen“

Zu guter Letzt sollten Sie sicherstellen, dass der gewünschte Dienst auf den richtigen Runlevels aktiviert ist. Weitere Informationen darüber finden Sie in Abschnitt 2.1, „Konfigurieren von Diensten auf dem LVS-Router“.

3.6. Speichern der Netzwerkpaketfiltereinstellungen

Nachdem Sie die Netzwerkpaketfilter für Ihre Anforderungen konfiguriert haben, speichern Sie die Einstellungen, sodass diese nach einem Neustart wiederhergestellt werden. Geben Sie für **iptables** den folgenden Befehl ein:

```
/sbin/service iptables save
```

Dies speichert die Einstellungen in **/etc/sysconfig/iptables**, damit diese beim Systemstart wieder abgerufen werden können.

Sobald diese Datei geschrieben wurde, können Sie den Befehl **/sbin/service** zum Starten, Stoppen und Prüfen des Status (mittels status-Switch) von **iptables** verwenden. Der Befehl **/sbin/service** lädt automatisch das richtige Modul für Sie. Ein Anwendungsbeispiel für **/sbin/service** finden Sie in Abschnitt 2.3, „Starten des **Piranha-Konfigurationstool**-Dienstes“.

Zu guter Letzt sollten Sie sicherstellen, dass der gewünschte Dienst auf den richtigen Runlevels aktiviert ist. Weitere Informationen darüber finden Sie in Abschnitt 2.1, „Konfigurieren von Diensten auf dem LVS-Router“.

Das nächste Kapitel erläutert die Verwendung des **Piranha-Konfigurationstools** zur Konfiguration des LVS-Routers und beschreibt die notwendigen Schritte zur Aktivierung des Load Balancer Add-Ons.

Kapitel 4. Konfigurieren des Load Balancer Add-Ons mit dem Piranha-Konfigurationstool

Das **Piranha-Konfigurationstool** bietet eine strukturierte Herangehensweise zum Erstellen der notwendigen Konfigurationsdatei für das Load Balancer Add-On: **/etc/sysconfig/ha/lvs.cf**. Dieses Kapitel beschreibt die grundlegende Verwendung des **Piranha-Konfigurationstools** und die Aktivierung des Load Balancer Add-Ons nach abgeschlossener Konfiguration.

Wichtig

Die Konfigurationsdatei für das Load Balancer Add-On folgt strengen Formatierungsregeln. Mithilfe des **Piranha-Konfigurationstools** lassen sich Syntaxfehler in der Datei **lvs.cf** und somit Softwarefehler vermeiden.

4.1. Notwendige Software

Der Dienst **piranha-gui** muss auf dem primären LVS-Router laufen, um das **Piranha-Konfigurationstool** verwenden zu können. Für die Konfiguration des Load Balancer Add-Ons brauchen Sie mindestens einen textbasierten Webbrowser wie z. B. **links**. Falls Sie von einem anderen Rechner aus auf den LVS-Router zugreifen, benötigen Sie zudem eine **ssh**-Verbindung zum primären LVS-Router als root-Benutzer.

Während der Konfiguration des primären LVS-Routers empfiehlt es sich, eine laufende **ssh**-Verbindung in einem Terminalfenster zu haben. Diese Verbindung bietet einen sicheren Weg zum Neustarten von **pulse** und anderen Diensten, zur Konfiguration von Netzwerkpaketfiltern und zur Überwachung von **/var/log/messages** während der Suche und Bereinigung von Fehlern.

Die nächsten vier Abschnitte führen Sie durch alle Konfigurationsseiten des **Piranha-Konfigurationstools** und liefern Anweisungen zur Einrichtung des Load Balancer Add-Ons.

4.2. Anmelden beim Piranha-Konfigurationstool

Bei der Konfiguration des Load Balancer Add-Ons sollten Sie immer damit beginnen, den primären Router mit dem **Piranha-Konfigurationstool** zu konfigurieren. Überprüfen Sie dazu, dass der Dienst **piranha-gui** läuft und dass das Administratorpasswort festgelegt wurde, wie in Abschnitt 2.2, „Einrichten eines Passworts für das **Piranha-Konfigurationstool**“ beschrieben.

Falls Sie lokal auf den Rechner zugreifen, können Sie **http://localhost:3636** in einem Webbrowser aufrufen, um das **Piranha-Konfigurationstool** zu öffnen. Geben Sie andernfalls den Hostnamen oder die reale IP-Adresse für den Server ein, gefolgt von **:3636**. Sobald der Browser die Verbindung hergestellt hat, sehen Sie den Bildschirm aus Abbildung 4.1, „Der Begrüßungsbildschirm“.

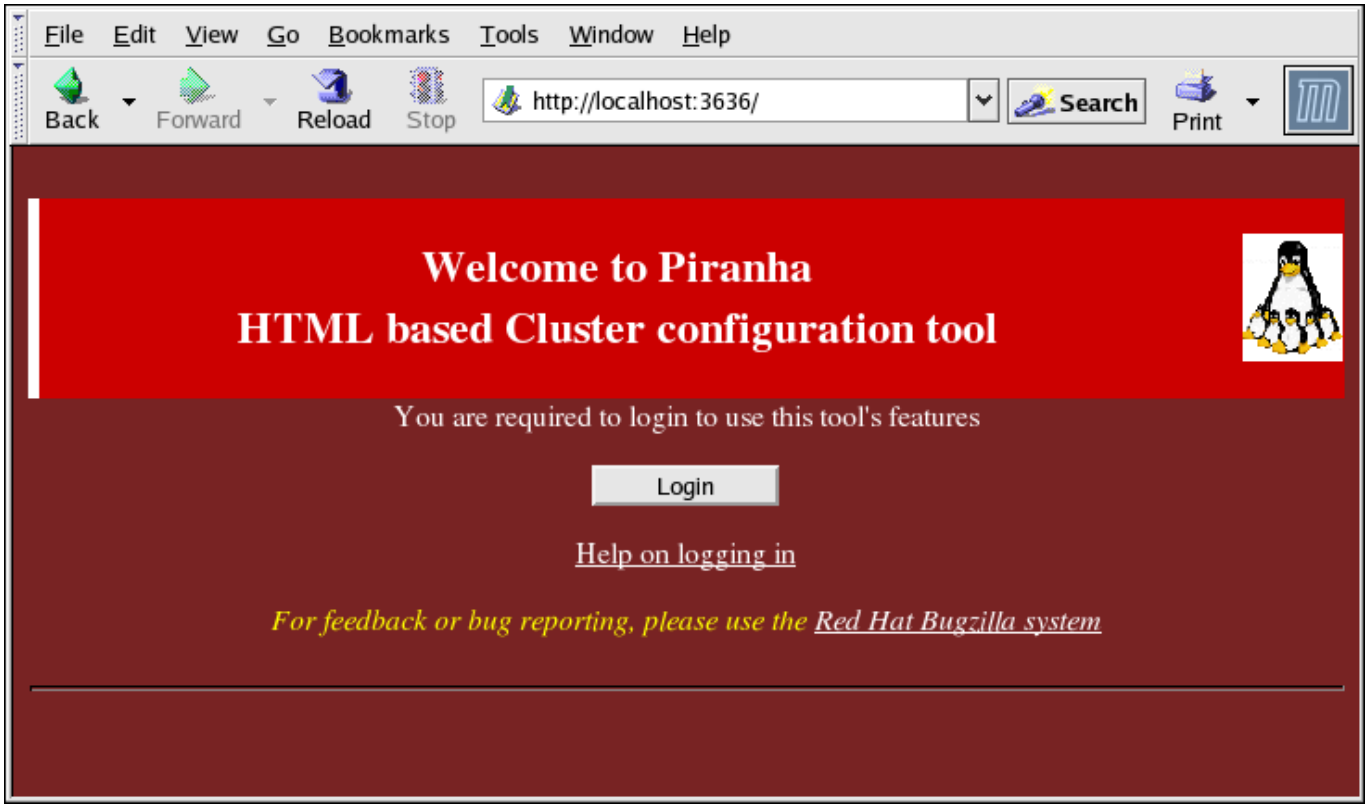

File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:3636/
Search
Print
Welcome to Piranha
HTML based Cluster configuration tool
You are required to login to use this tool's features
Login
Help on logging in
For feedback or bug reporting, please use the Red Hat Bugzilla system

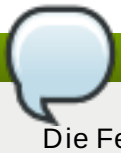
Abbildung 4.1. Der Begrüßungsbildschirm

Klicken Sie auf die Schaltfläche **Login** und geben Sie **piranha** als **Username** und das von Ihnen festgelegte Administratorpasswort im Feld **Password** ein.

Das **Piranha-Konfigurationstool** besteht im Wesentlichen aus vier Bildschirmen oder *Reitern*. Darüber hinaus enthält der Reiter **Virtual Servers** vier *Unterbereiche*. Nach der Anmeldung sehen Sie als Erstes den Reiter **CONTROL/MONITORING**.

4.3. CONTROL/MONITORING

Der Reiter **CONTROL/MONITORING** zeigt einen begrenzten Laufzeitstatus des Load Balancer Add-Ons an. Hier wird der Status des **pulse**-Daemons, der LVS-Routing-Tabelle und der von LVS erzeugten **nanny**-Prozesse angezeigt.



Anmerkung

Die Felder für **CURRENT LVS ROUTING TABLE** und **CURRENT LVS PROCESSES** bleiben leer, bis Sie das Load Balancer Add-On starten, wie in Abschnitt 4.8, „Starten des Load Balancer Add-Ons“ gezeigt.

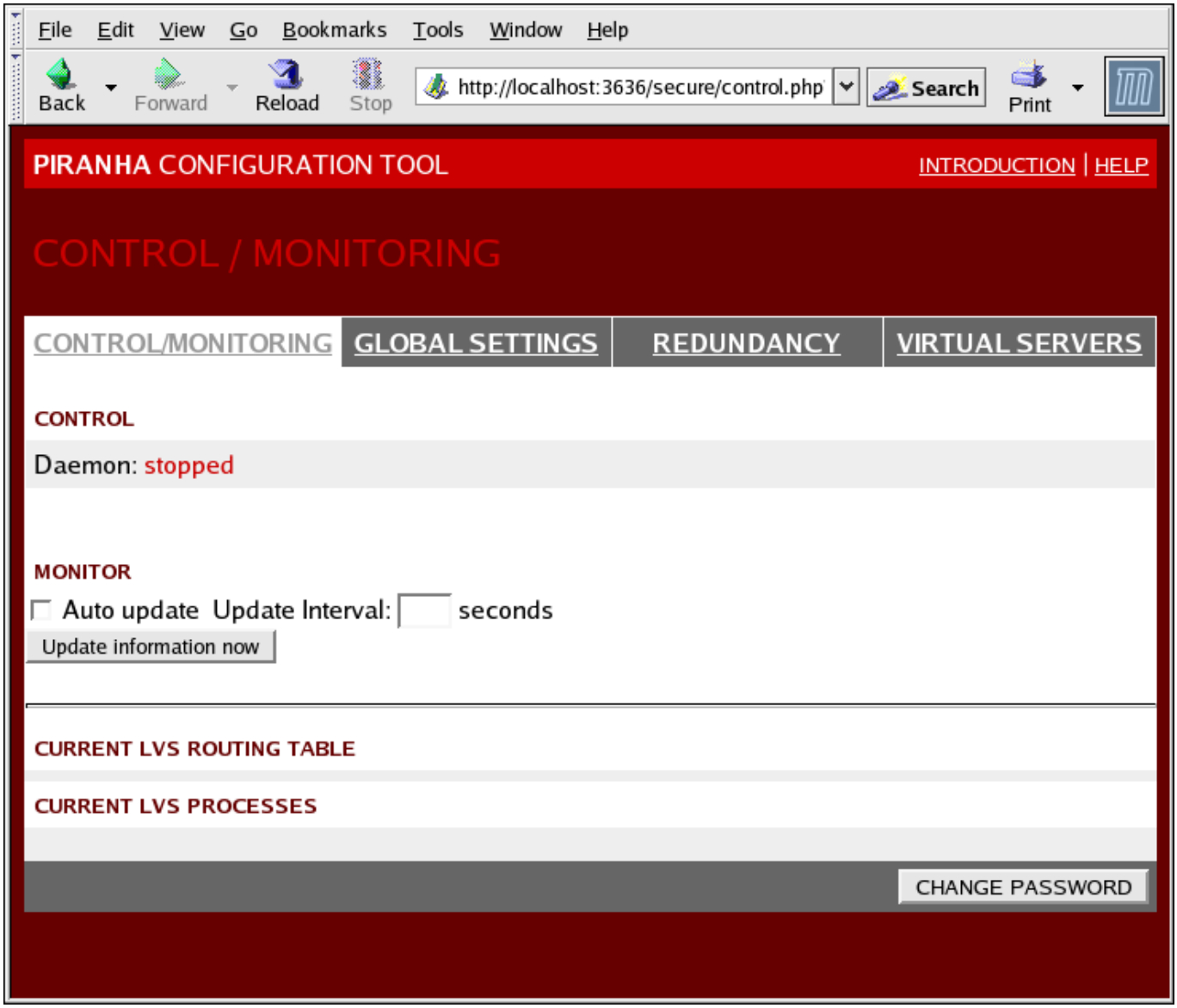


Abbildung 4.2. Der Reiter `CONTROL/MONITORING`

`Auto update`

Die Statusanzeige auf dieser Seite kann automatisch aktualisiert werden in benutzerdefinierten Zeitabständen. Um diese Funktion zu aktivieren, markieren Sie das Auswahlkästchen **`Auto update`** und geben Sie das gewünschte Intervall im Textfeld **`Update frequency in seconds`** ein (der Standardwert beträgt 10 Sekunden).

Es wird nicht empfohlen, die automatische Aktualisierung auf einen Zeitintervall von weniger als 10 Sekunden zu setzen. Andernfalls kann es problematisch werden, das Zeitintervall für **`Auto update`** neu zu konfigurieren, da die Seite zu schnell neu lädt. Falls Sie dieses Problem haben, klicken Sie einfach auf einen anderen Reiter und dann zurück auf **`CONTROL/MONITORING`**.

Die Funktion **`Auto update`** funktioniert nicht mit allen Browsern wie z. B. **`Mozilla`**.

`Update information now`

Sie können die Statusinformationen auch manuell aktualisieren, indem Sie auf diese Schaltfläche klicken.

`CHANGE PASSWORD`

Ein Klick auf diese Schaltfläche führt Sie zu einer Hilfeseite mit Informationen, wie Sie das administrative Passwort für das **Piranha-Konfigurationstool** ändern können.

4.4. GLOBAL SETTINGS

Auf dem Reiter **GLOBAL SETTINGS** definieren Sie die Netzwerkdetails für die öffentliche und die private Schnittstelle des primären LVS-Routers.

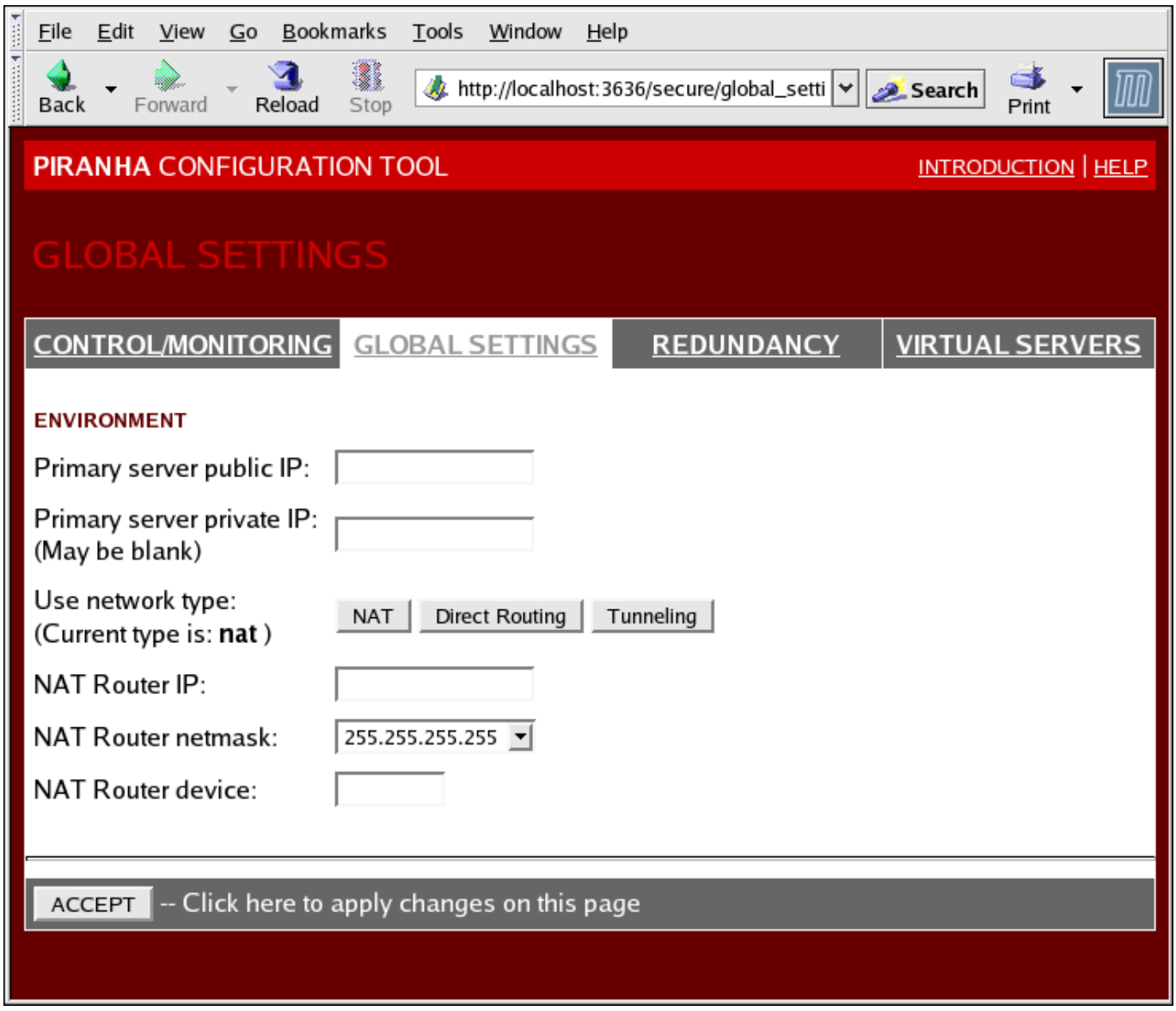


Abbildung 4.3. Der Reiter GLOBAL SETTINGS

Die obere Hälfte dieses Reiters konfiguriert die öffentliche und die private Netzwerkschnittstelle des primären LVS-Routers. Diese Schnittstellen wurden bereits in Abschnitt 3.1.1, „Konfigurieren von Netzwerkschnittstellen für Load Balancer Add-On mit NAT“ konfiguriert.

Primary server public IP

Geben Sie in diesem Feld die öffentlich routbare, reale IP-Adresse für den primären LVS-Knoten ein.

Primary server private IP

Geben Sie in diesem Feld die reale IP-Adresse für eine alternative Netzwerkschnittstelle auf dem primären LVS-Knoten ein. Diese Adresse wird lediglich als alternativer Kanal zur Übertragung von Heartbeats für den Backup-Router verwendet und muss nicht mit der realen privaten IP-Adresse übereinstimmen, die in Abschnitt 3.1.1, „Konfigurieren von Netzwerkschnittstellen für Load Balancer Add-On mit NAT“ zugewiesen wird. Sie können dieses Feld frei lassen, sollten sich jedoch darüber im Klaren sein, dass es in diesem Fall keinen alternativen Heartbeat-Kanal für den Backup-LVS-Router gibt und dies einen Single Point of Failure darstellt.

Anmerkung

Die private IP-Adresse ist nicht notwendig für Konfigurationen mit **Direct Routing**, da alle realen Server und die LVS-Router dieselbe virtuelle IP-Adresse verwenden und dieselbe IP-Route konfiguriert haben sollten.

Anmerkung

Die private IP-Adresse des primären LVS-Routers kann auf jeder Schnittstelle konfiguriert werden, die TCP/IP akzeptiert, egal ob Ethernet-Adapter oder serieller Port.

TCP Timeout

Geben Sie die Zeit (in Sekunden) an, nach der für eine TCP-Sitzung eine Zeitüberschreitung erfolgt. Der Standardwert ist 0.

TCP Fin Timeout

Geben Sie die Zeit (in Sekunden) an, nach der für eine TCP-Sitzung nach Empfang eines FIN-Pakets eine Zeitüberschreitung erfolgt. Der Standardwert ist 0.

UDP Timeout

Geben Sie die Zeit (in Sekunden) an, nach der für eine UDP-Sitzung eine Zeitüberschreitung erfolgt. Der Standardwert ist 0.

Use network type

Klicken Sie auf die Schaltfläche **NAT**, um das NAT-Routing auszuwählen.

Klicken Sie auf die Schaltfläche **Direct Routing**, um das direkte Routing auszuwählen.

Die nächsten drei Felder beziehen sich auf die virtuelle Netzwerkschnittstelle des NAT-Routers, die das private Netzwerk mit den realen Servern verbinden. Diese Felder haben in einer Konfiguration mit direktem Routing *keine* Auswirkung.

NAT Router IP

Geben Sie die private Floating-IP in diesem Textfeld ein. Diese Floating-IP sollte als Gateway für die realen Server verwendet werden.

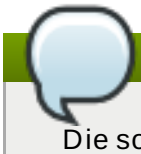
NAT Router netmask

Falls die Floating-IP des NAT-Routers eine spezielle Netzmaske benötigt, wählen Sie diese

aus der Auswahlliste.

NAT Router device

Geben Sie in diesem Textfeld den Gerätenamen der Netzwerkschnittstelle für die Floating-IP-Adresse an, z. B. **eth1:1**.



Anmerkung

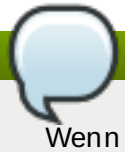
Die sollten ein Alias für die Floating-IP-Adresse von NAT zur Ethernet-Schnittstelle, die mit dem privaten Netzwerk verbunden ist, anlegen. In diesem Beispiel ist das private Netzwerk auf der Schnittstelle **eth1**, somit ist **eth1:1** die Floating-IP-Adresse.

Warnung

Sobald Sie diese Seite vervollständigt haben, klicken Sie auf die Schaltfläche **ACCEPT**, um sicherzustellen, dass keine Änderungen verloren gehen, wenn Sie den nächsten Reiter anklicken.

4.5. REDUNDANCY

Der Reiter **REDUNDANCY** ermöglicht Ihnen die Konfiguration des Backup-LVS-Routerknotens und das Einstellen verschiedener Heartbeat-Überwachungsoptionen.



Anmerkung

Wenn Sie diesen Bildschirm zum ersten Mal aufrufen, wird der **Backup**-Status „inactive“ angezeigt sowie die Schaltfläche **ENABLE**. Um den Backup-LVS-Router zu konfigurieren, klicken Sie auf die Schaltfläche **ENABLE**, damit der Bildschirm aussieht wie in Abbildung 4.4, „Der Reiter **REDUNDANCY**“ dargestellt.

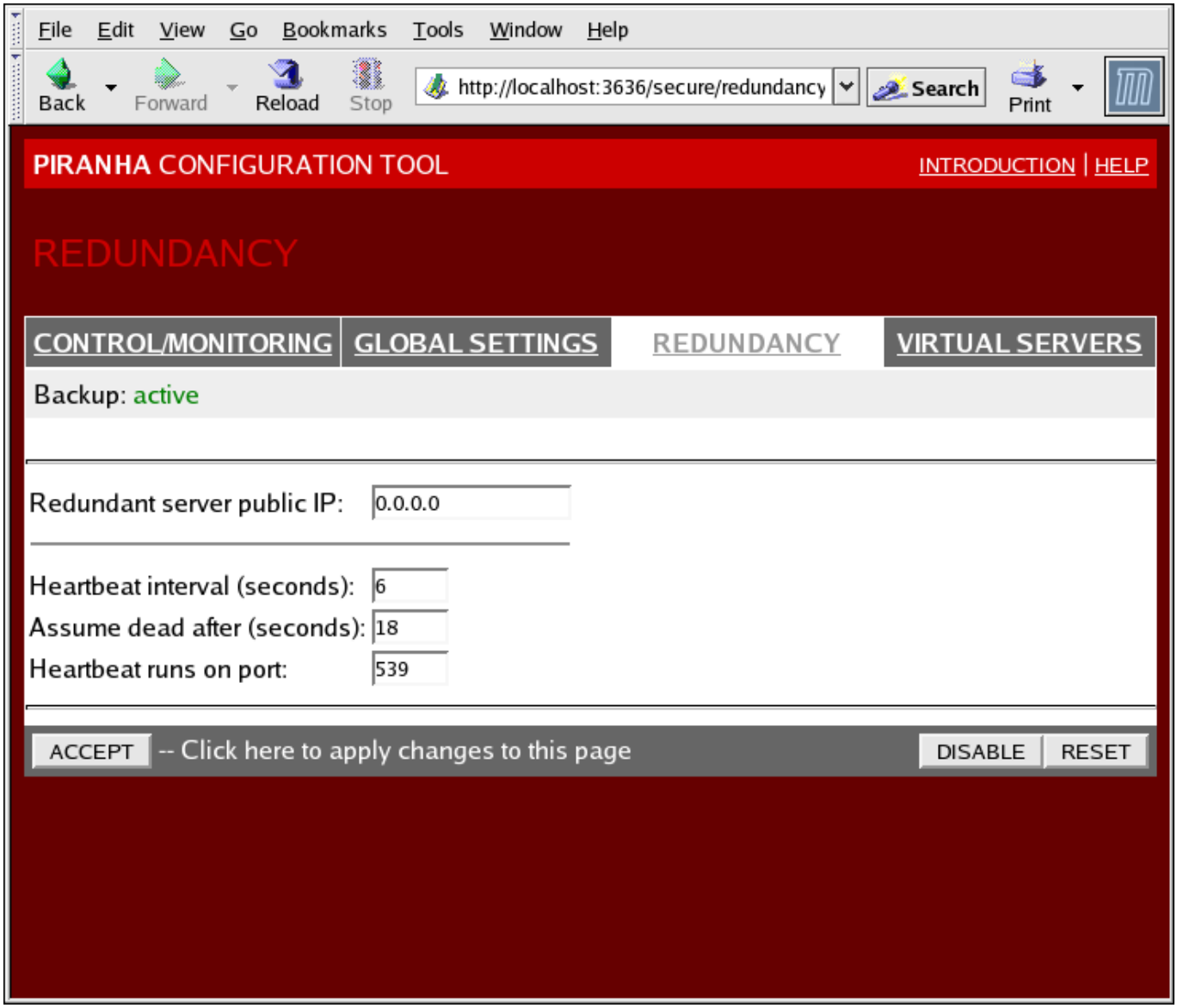


Abbildung 4.4. Der Reiter REDUNDANCY

Redundant server public IP

Geben Sie die öffentliche, reale IP-Adresse für den Backup-LVS-Routerknoten ein.

Redundant server private IP

Geben Sie in diesem Textfeld die private, reale IP-Adresse des Backup-Knotens ein.

Falls Sie das Feld **Redundant server private IP** nicht sehen, gehen Sie zurück zum Reiter **GLOBAL SETTINGS**, geben Sie eine IP-Adresse für **Primary server private IP** an und klicken Sie auf **ACCEPT**.

Der nächste Bereich des Reiters dient zur Konfiguration des Heartbeat-Channels, der vom Backup-Knoten verwendet wird, um den primären Knoten auf einen möglichen Ausfall hin zu überwachen.

Heartbeat Interval (seconds)

Dieses Feld legt die Anzahl an Sekunden zwischen Heartbeats fest, also die Zeitabstände, in denen der Backup-Knoten den funktionalen Status des primären LVS-Knoten überprüft.

Assume dead after (seconds)

Falls der primäre LVS-Knoten nicht nach dieser Anzahl an Sekunden antwortet, leitet der Backup-LVS-Knoten den Failover ein.

Heartbeat runs on port

Dieses Feld legt den Port fest, auf dem der Heartbeat mit dem primären LVS-Knoten kommuniziert. Der Standardwert ist 539, falls dieses Feld leer gelassen wird.

Der letzte Bereich dieses Reiters dient der Aktivierung und Konfiguration des optionalen Synchronisations-Daemons und dessen Optionen. Der Synchronisations-Daemon aktiviert den aktiven und den Backup-LVS-Router, um den TCP-Status synchron zu halten. Wenn aktiviert, sendet der aktive Router eine Multicast-Nachricht mit einer konfigurierbaren Synchronisations-ID (oder syncid) über das Netzwerk an einen empfangenden Backup-Router.

Warnung

Red Hat Enterprise Linux 6.5 führt ein neues Protokollformat für Synchronisationsnachrichten ein, das Unterbrechungen bei Geschäftsdiensten verhindern soll, die durch eine vorzeitige Zeitüberschreitung persistenter Verbindungen auf Backup-Knoten verursacht werden und so inkonsistente Zustände beim Failover zur Folge haben.

Das neue Protokollformat ist nicht kompatibel mit Red Hat Enterprise Linux 6.4 oder früheren Versionen oder Kernel-Versionen vor kernel-2.6.32-406.el6. Es wird empfohlen, Backup-Knoten zu aktualisieren, bevor der Master-Knoten auf Red Hat Enterprise Linux 6.5 aktualisiert wird.

Um weiterhin das alte Format für Synchronisationsnachrichten zu verwenden (falls Sie beispielsweise den Master-Knoten vor den Backup-Knoten synchronisieren müssen), setzen Sie den Wert für **sync_version** mithilfe des **echo**-Befehls als root-Benutzer an einer Shell-Eingabeaufforderung wie folgt:

```
echo 0 > /proc/sys/net/ipv4/vs/sync_version
```

Use Sync Daemon

Markieren Sie das Auswahlkästchen, falls Sie den Synchronisations-Daemon aktivieren möchten.

Sync Daemon Interface

Die Netzwerkschnittstelle, über die der Synchronisations-Daemon seine Multicast-Nachrichten sendet bzw. empfängt. Die standardmäßige Schnittstelle in diesem Feld ist **eth0**.

Sync daemon id

Dieses Feld legt eine Kennung (ID) für Multicast-Synchronisationsnachrichten fest. Unterstützte Werte sind 0 bis 255. Der Standardwert ist 0, falls dieses Feld leer gelassen wird.

Warnung

Vergessen Sie nicht, auf die Schaltfläche **ACCEPT** zu klicken, um sicherzustellen, dass keine Änderungen verloren gehen, wenn Sie den nächsten Reiter anklicken.

4.6. VIRTUAL SERVERS

Der Reiter **VIRTUAL SERVERS** zeigt Informationen zu jedem derzeit definierten virtuellen Server an. Jeder Tabelleneintrag zeigt den Status des virtuellen Servers, dessen Namen, die dem Server zugeteilte virtuelle IP, die Netzmaske der virtuellen IP, die Portnummer, auf der der Dienst kommuniziert, das verwendete Protokoll und die Schnittstelle des virtuellen Geräts.

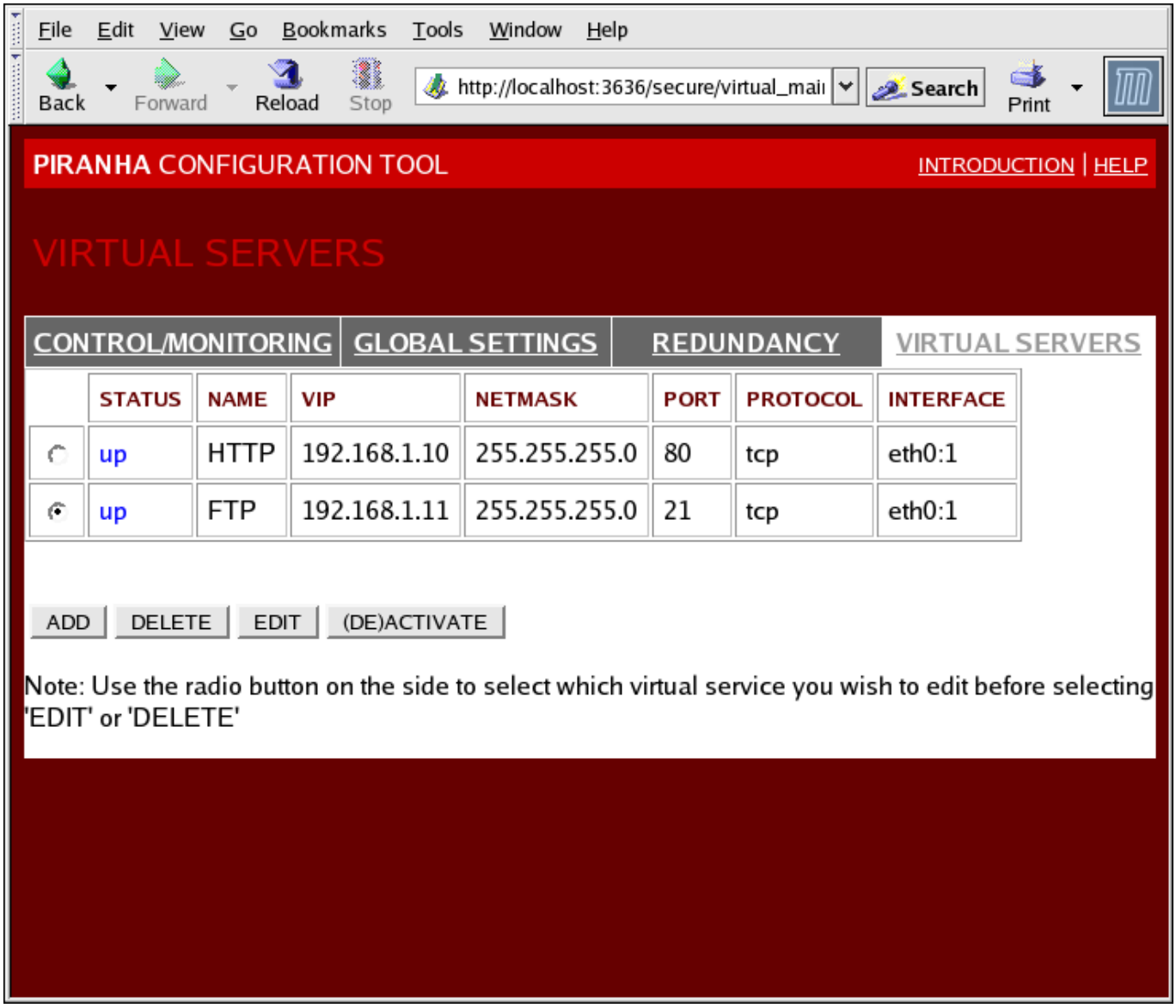

File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:3636/secure/virtual_mai
Search
Print
PIRANHA CONFIGURATION TOOL
INTRODUCTION | HELP
VIRTUAL SERVERS
CONTROL/MONITORING
GLOBAL SETTINGS
REDUNDANCY
VIRTUAL SERVERS
STATUS NAME VIP NETMASK PORT PROTOCOL INTERFACE
up HTTP 192.168.1.10 255.255.255.0 80 tcp eth0:1
up FTP 192.168.1.11 255.255.255.0 21 tcp eth0:1
ADD DELETE EDIT (DE)ACTIVATE
Note: Use the radio button on the side to select which virtual service you wish to edit before selecting 'EDIT' or 'DELETE'

Abbildung 4.5. Der Reiter VIRTUAL SERVERS

Jeder Server, der im Reiter **VIRTUAL SERVERS** angezeigt wird, kann auf den nachfolgenden Bildschirmen oder *Unterabschnitten* konfiguriert werden.

Um einen Dienst hinzuzufügen, klicken Sie auf die Schaltfläche **ADD**. Um einen Dienst zu entfernen, wählen Sie diesen aus, indem Sie auf die Auswahlfläche neben dem virtuellen Server und dann auf die Schaltfläche **DELETE** klicken.

Um einen virtuellen Server in der Tabelle zu aktivieren oder zu deaktivieren, klicken Sie auf dessen Auswahlfläche und anschließend auf die Schaltfläche **(DE)ACTIVATE**.

Nach Hinzufügen eines virtuellen Servers können Sie diesen konfigurieren, indem Sie auf die Auswahlfläche links daneben und anschließend auf die Schaltfläche **EDIT** klicken, um den Unterabschnitt **VIRTUAL SERVER** anzuzeigen.

4.6.1. Der Unterabschnitt VIRTUAL SERVER

Der Unterabschnitt **VIRTUAL SERVER**, wie in Abbildung 4.6, „Der Unterabschnitt **VIRTUAL SERVERS**“ gezeigt, ermöglicht Ihnen die Konfiguration eines einzelnen virtuellen Servers. Links zu Unterabschnitten, die sich speziell auf diesen virtuellen Server beziehen, befinden sich entlang des oberen Seitenrands. Bevor Sie jedoch damit beginnen, die Unterabschnitte für diesen virtuellen Server zu konfigurieren, vervollständigen Sie diese Seite und klicken Sie auf **ACCEPT**.

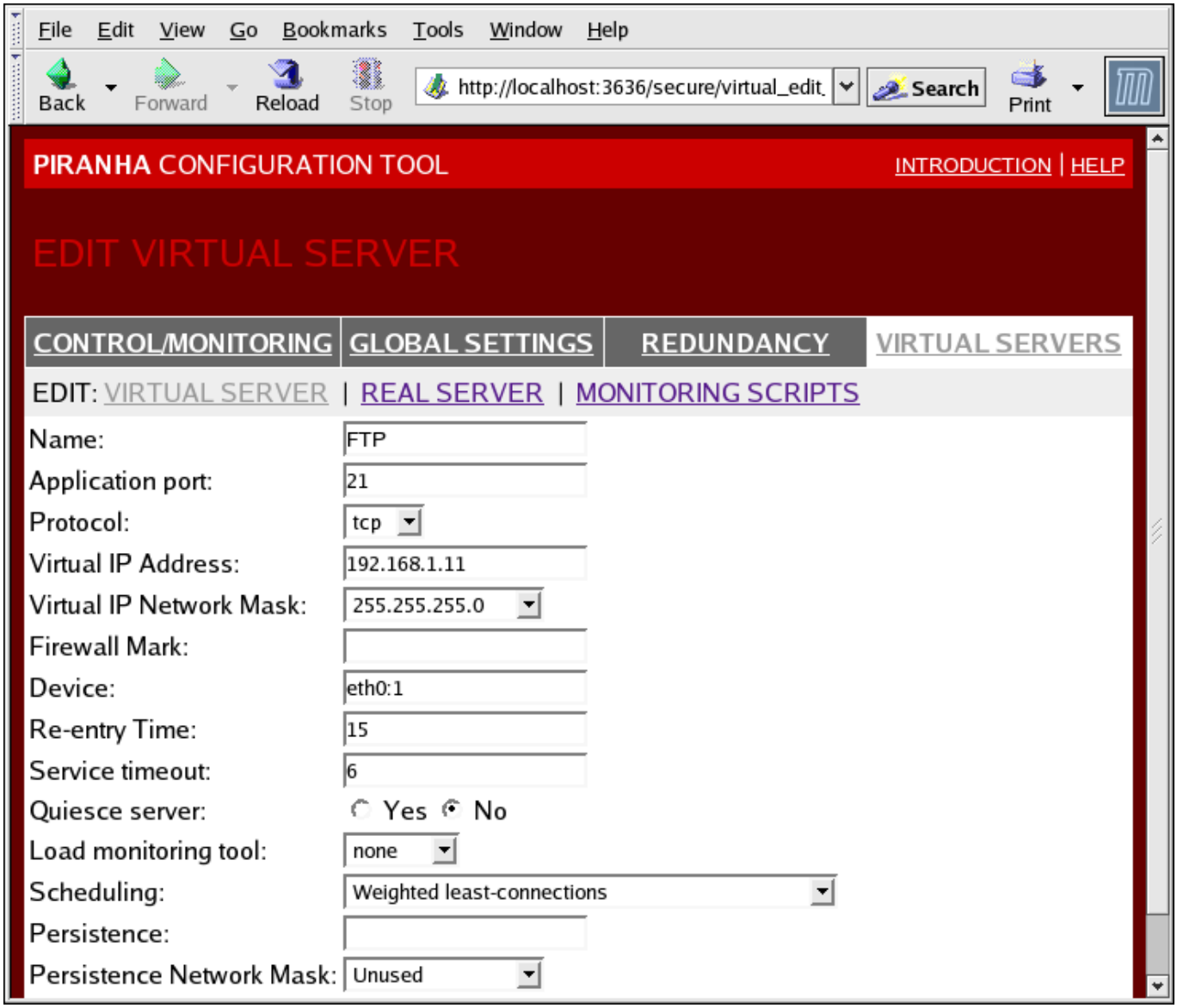


Abbildung 4.6. Der Unterabschnitt VIRTUAL SERVERS

Name

Geben Sie einen aussagekräftigen Namen zur Identifizierung des virtuellen Servers ein. Dieser Name ist *nicht* der Hostname für den Rechner und sollte daher aussagekräftig und einfach identifizierbar sein. Sie können sogar das Protokoll angeben, das vom virtuellen Server verwendet werden soll, wie zum Beispiel HTTP.

Application port

Geben Sie die Portnummer an, auf der die Dienstapplikation lauschen soll. Da dieses Beispiel sich auf HTTP-Dienste bezieht, wird der Port 80 verwendet.

Protocol

Wählen Sie aus der Auswahlliste zwischen UDP und TCP. Webserver kommunizieren in der Regel über das TCP-Protokoll, weshalb im obigen Beispiel TCP ausgewählt wird.

Virtual IP Address

Geben Sie in diesem Textfeld die Floating-IP-Adresse des virtuellen Servers ein.

Virtual IP Network Mask

Wählen Sie aus der Auswahlliste die Netzmaske für diesen virtuellen Server.

Firewall Mark

Geben Sie in diesem Feld *keinen* Ganzzahlwert für die Firewall-Markierung ein, es sei denn, Sie bündeln Protokolle mit mehreren Ports oder Sie erstellen einen virtuellen Server mit mehreren Ports für separate, aber verwandte Protokolle. In diesem Beispiel hat der obige virtuelle Server die **Firewall Mark** 80, um Verbindungen mit HTTP auf Port 80 und mit HTTPS auf Port 443 mithilfe der Firewall-Markierung 80 zu bündeln. In Kombination mit Persistenz gewährleistet dieses Verfahren, dass Benutzer, die sowohl auf sichere als auch auf unsichere Websites zugreifen, an denselben Server geleitet werden, um so den Zustand zu bewahren.

Warnung

Die Eingabe einer Firewall-Markierung in diesem Feld informiert IPVS darüber, dass Pakete mit dieser Firewall-Markierung gleich behandelt werden sollen. Um diese Firewall-Markierung zuzuweisen, sind jedoch zusätzliche Konfigurationsschritte außerhalb des **Piranha-Konfigurationstools** notwendig. Siehe Abschnitt 3.4, „Multi-Port-Dienste und Load Balancer Add-On“ für Anweisungen zur Erstellung von Multi-Port-Diensten und Abschnitt 3.5, „Konfigurieren von FTP“ für Anweisungen zur Erstellung eines hochverfügbaren, virtuellen FTP-Servers.

Device

Geben Sie den Namen des Netzwerkgeräts ein, mit dem die im Feld **Virtual IP Address** definierte Floating-IP-Adresse verknüpft werden soll.

Sie sollten ein Alias anlegen für die öffentliche Floating-IP-Adresse zur Ethernet-Schnittstelle, die mit dem öffentlichen Netzwerk verbunden ist. In diesem Beispiel ist das öffentliche Netzwerk auf der Schnittstelle **eth0**, somit sollte **eth0:1** als Gerätename eingegeben werden.

Re-entry Time

Geben Sie einen ganzzahligen Wert ein, der die Zeitspanne in Sekunden definiert, bevor der aktive LVS-Router versucht, einen realen Server im Anschluss an einen Ausfall wieder in den Pool einzubringen.

`Service Timeout`

Geben Sie einen ganzzahligen Wert an, der die Zeitspanne in Sekunden definiert, bevor ein realer Server als ausgefallen gilt und aus dem Pool entfernt wird.

`Quiesce server`

Wenn die Auswahlfläche **`Quiesce server`** markiert ist, wird die Gewichtung eines realen Servers auf 0 gesetzt, wenn dieser nicht verfügbar ist. Dadurch wird der Server effektiv deaktiviert. Wenn der reale Server wieder verfügbar ist, wird er wieder aktiviert, indem seine ursprüngliche Gewichtung wiederhergestellt wird. Falls **`Quiesce server`** deaktiviert ist, wird ein ausgefallener Server aus der Servertabelle entfernt. Wenn dieser ausgefallene reale Server wieder verfügbar wird, dann wird er wieder zur Tabelle der realen Server hinzugefügt.

`Load monitoring tool`

Der LVS-Router kann die Auslastung auf verschiedenen realen Servern entweder mithilfe von **`rup`** oder **`ruptime`** überwachen. Falls Sie **`rup`** aus dem Auswahlmenü auswählen, muss auf jedem realen Server der **`rstatd`**-Dienst laufen. Falls Sie **`ruptime`** auswählen, muss auf jedem realen Server der **`rwhod`**-Dienst laufen.

Warnung

Die Überwachung der Auslastung ist *nicht* dasselbe wie die Lastverteilung und kann zu schwer vorhersehbarem Scheduling-Verhalten führen, falls sie mit gewichteten Scheduling-Algorithmen kombiniert wird. Zudem muss es sich zur Überwachung der Auslastung bei den realen Servern um Linux-Rechner handeln.

`Scheduling`

Wählen Sie Ihren bevorzugten Scheduling-Algorithmus aus der Auswahlliste. Der Standard ist **`Weighted least-connection`**. Weitere Informationen über Scheduling-Algorithmen finden Sie in Abschnitt 1.3.1, „Scheduling-Algorithmen“.

`Persistence`

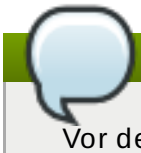
Falls Sie persistente Verbindungen mit dem virtuellen Server während Client-Transaktionen benötigen, geben Sie in diesem Textfeld die Anzahl der Sekunden für Inaktivität an, bevor eine Zeitüberschreitung der Verbindung erfolgt.

Wichtig

Falls Sie einen Wert in dem Feld **`Firewall Mark`** oben eingegeben haben, sollten Sie ebenfalls einen Wert für die Persistenz eingeben. Vergewissern Sie sich zudem, dass Sie bei der Verwendung von Firewall-Markierungen mit Persistenz für jeden virtuellen Server mit der Firewall-Markierung dieselben Werte für die Persistenz angeben. Weitere Informationen über Persistenz und Firewall-Markierungen finden Sie in Abschnitt 1.5, „Persistenz und Firewall-Markierungen“.

Persistence Network Mask

Um die Persistenz für ein bestimmtes Subnetz einzuschränken, wählen Sie die entsprechende Netzwerkmaske aus der Auswahlliste.



Anmerkung

Vor dem Aufkommen der Firewall-Markierungen war die auf Subnetze eingeschränkte Persistenz eine grobe Methode zur Bündelung von Verbindungen. Heutzutage ist es am besten, zu diesem Zweck Persistenz zusammen mit Firewall-Markierungen zu verwenden.

Warnung

Vergessen Sie nicht, auf die Schaltfläche **ACCEPT** zu klicken, um sicherzustellen, dass keine Änderungen verloren gehen, wenn Sie den nächsten Reiter anklicken.

4.6.2. Der Unterabschnitt REAL SERVER

Ein Klick auf den Link zum Unterabschnitt **REAL SERVER** im oberen Bereich des Reiters zeigt den Unterabschnitt **EDIT REAL SERVER** an. Hier wird der Status des physischen Server-Hosts für einen bestimmten virtuellen Dienst angezeigt.

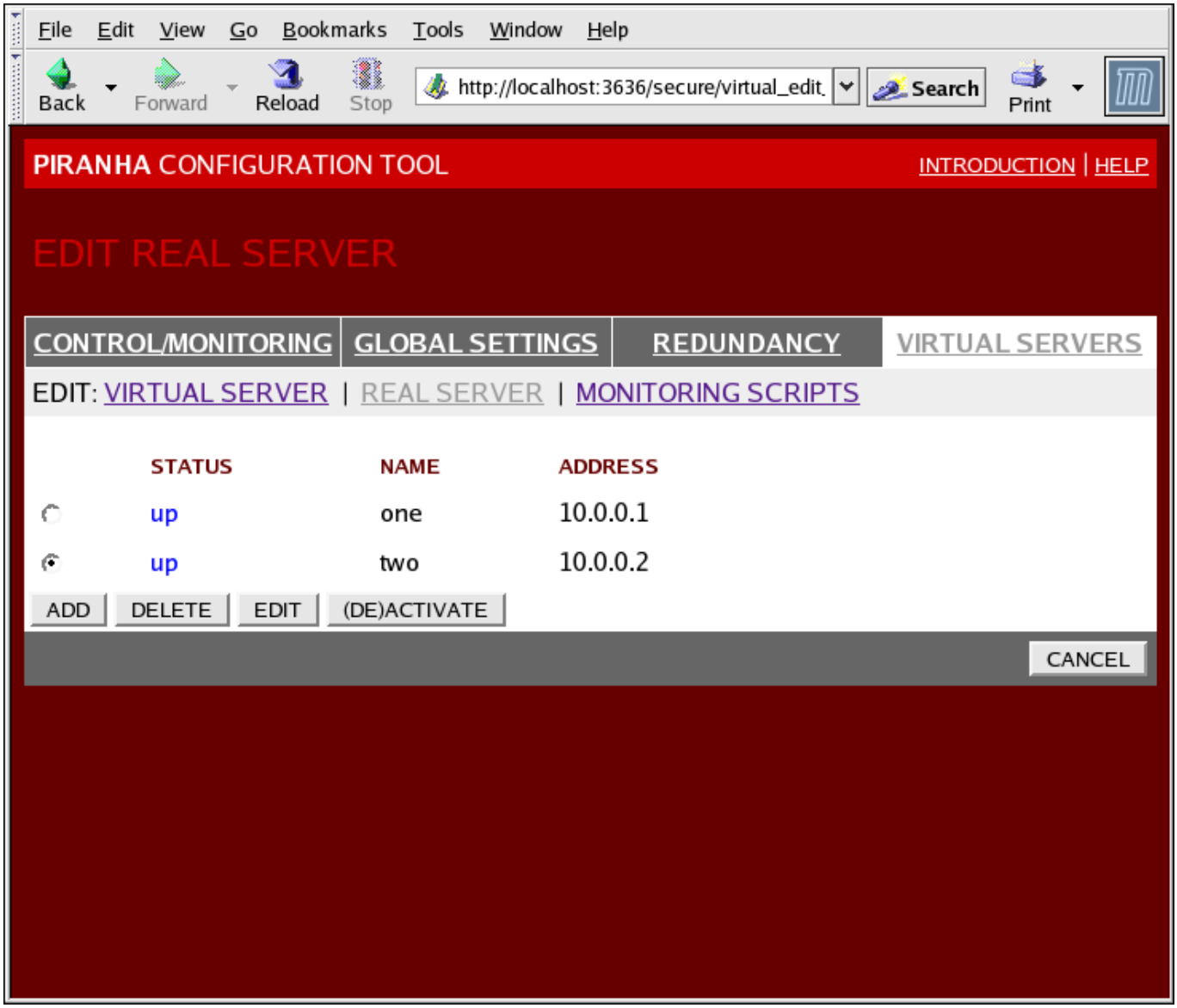


Abbildung 4.7. Der Unterabschnitt REAL SERVER

Klicken Sie auf die Schaltfläche **ADD**, um einen neuen Server hinzuzufügen. Um einen vorhandenen Server zu löschen, markieren Sie die Auswahlfläche neben dem zu löschenden Server und klicken Sie auf **DELETE**. Klicken Sie auf die Schaltfläche **EDIT**, um den Reiter **EDIT REAL SERVER** zu laden, wie in Abbildung 4.8, „Der Konfigurationsreiter **REAL SERVER**“ dargestellt.

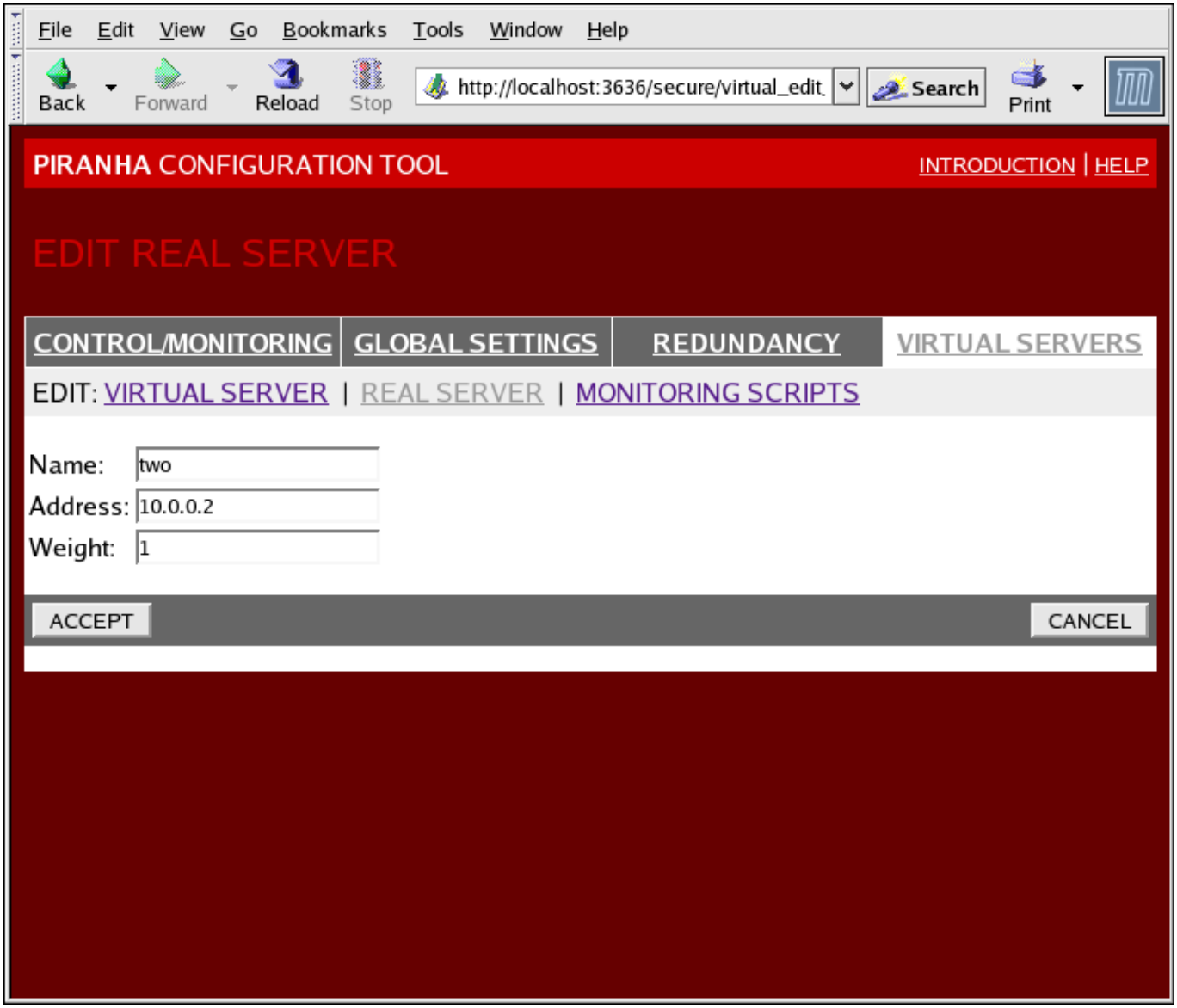
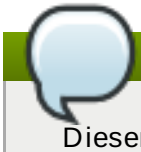


Abbildung 4.8. Der Konfigurationsreiter REAL SERVER

Dieser Reiter besteht aus drei Eingabefeldern:

Name

Ein beschreibender Name für den realen Server.



Anmerkung

Dieser Name ist *nicht* der Hostname für den Rechner und sollte daher aussagekräftig und einfach identifizierbar sein.

Address

Die IP-Adresse des realen Servers. Da der Port, auf dem gelauscht wird, bereits für den dazugehörigen virtuellen Server angegeben ist, fügen Sie hier keine Portnummer hinzu.

Weight

Ein ganzzahliger Wert, der die Kapazität dieses Hosts relativ zu der Kapazität anderer Hosts im Pool anzeigt. Der Wert kann beliebig sein, sollte jedoch als Faktor im Vergleich zu anderen realen Servern gewählt werden. Weitere Informationen über Servergewichtung finden Sie unter Abschnitt 1.3.2, „Server-Gewichtung und Scheduling“.

Warnung

Vergessen Sie nicht, auf die Schaltfläche **ACCEPT** zu klicken, um sicherzustellen, dass keine Änderungen verloren gehen, wenn Sie den nächsten Reiter anklicken.

4.6.3. Der Unterabschnitt EDIT MONITORING SCRIPTS

Klicken Sie auf den Link **MONITORING SCRIPTS** oben auf der Seite. Der Unterabschnitt **EDIT MONITORING SCRIPTS** erlaubt dem Administrator die Angabe einer Send/Expect-String-Sequenz, um zu verifizieren, dass der Dienst für den virtuellen Server auf jedem realen Server funktionsfähig ist. Es ist außerdem der Ort, an dem der Administrator angepasste Skripte angeben kann, um Dienste zu überprüfen, die dynamisch verändernde Daten erfordern.

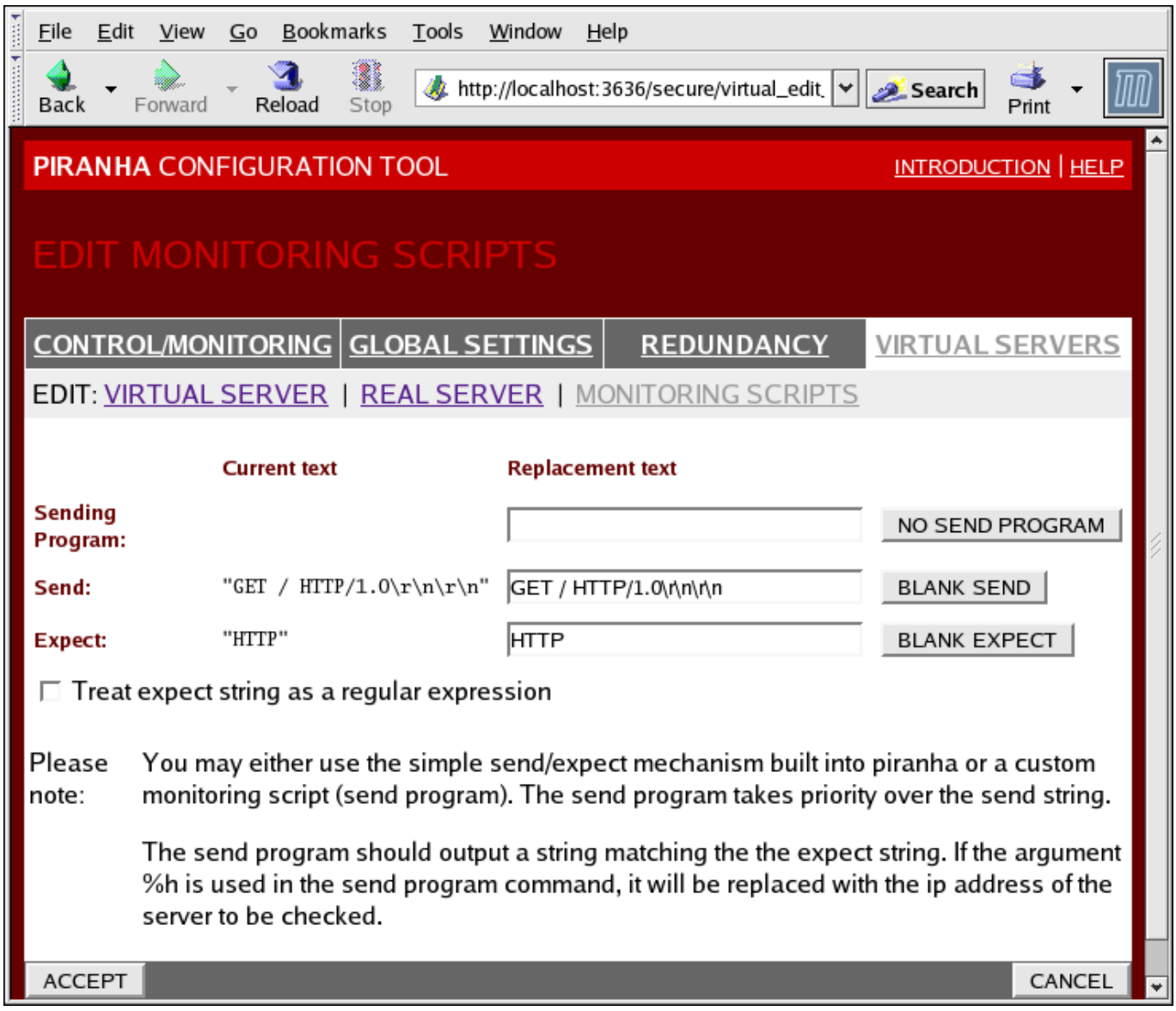

File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:3636/secure/virtual_edit.
Search
Print
PIRANHA CONFIGURATION TOOL
INTRODUCTION | HELP
EDIT MONITORING SCRIPTS
CONTROL/MONITORING
GLOBAL SETTINGS
REDUNDANCY
VIRTUAL SERVERS
EDIT: VIRTUAL SERVER | REAL SERVER | MONITORING SCRIPTS
Current text
Replacement text
Sending Program:
NO SEND PROGRAM
Send:
"GET / HTTP/1.0\r\n\r\n"
GET / HTTP/1.0\r\n\r\n
BLANK SEND
Expect:
"HTTP"
HTTP
BLANK EXPECT
Treat expect string as a regular expression
Please note:
You may either use the simple send/expect mechanism built into piranha or a custom monitoring script (send program). The send program takes priority over the send string.
The send program should output a string matching the the expect string. If the argument %h is used in the send program command, it will be replaced with the ip address of the server to be checked.
ACCEPT
CANCEL

Abbildung 4.9. Der Unterabschnitt EDIT MONITORING SCRIPTS

Sending Program

Für eine fortgeschrittenere Dienstverifizierung können Sie dieses Feld zur Angabe des Pfads zu einem Skript zur Dienstprüfung verwenden. Diese Funktion ist besonders für Dienste hilfreich, die dynamisch verändernde Daten erfordern, wie beispielsweise HTTPS oder SSL.

Um diese Funktionalität zu verwenden, müssen Sie ein Skript schreiben, das eine Antwort in Textform zurückgibt, dieses ausführbar machen und den Pfad im Feld **Sending Program** eingeben.

Anmerkung

Um sicherzustellen, dass jeder Server im Pool der realen Server geprüft wird, verwenden Sie einen speziellen Platzhalter **%h** nach dem Pfad zum Skript im Feld **Sending Program**. Dieser Platzhalter wird durch die IP-Adresse eines jeden realen Servers ersetzt, sobald das Skript vom **nanny**-Daemon aufgerufen wird.

Nachfolgend sehen Sie ein Beispielskript, das Ihnen als Leitfaden für die Erstellung Ihres Skripts zur Dienstprüfung dienen kann:

```
#!/bin/sh

TEST=`dig -t soa example.com @$1 | grep -c dns.example.com

if [ $TEST != "1" ]; then
 echo "OK
else
 echo "FAIL"
fi
```

Anmerkung

Falls im Feld **Sending Program** ein externes Programm eingegeben wird, dann wird das Feld **Send** ignoriert.

Send

Geben Sie in diesem Feld einen String für den **nanny**-Daemon für das Versenden an jeden realen Server ein. Standardmäßig ist das Feld **Send** für HTTP bereits vervollständigt. Sie können diesen Wert gemäß Ihrer Anforderungen anpassen. Falls Sie dieses Feld leer lassen, versucht der **nanny**-Daemon, den Port zu öffnen, und geht davon aus, dass der Dienst läuft, wenn er erfolgreich ist.

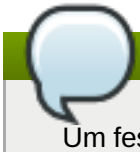
Nur eine Sendesequenz ist in diesem Feld gestattet und es darf nur druckbare ASCII-Zeichen sowie die folgenden Escape-Zeichen enthalten:

- \n für Zeilenumbruch
- \r für Wagenrücklauf
- \t für Tabulator

- \ für den Escape des darauffolgenden Zeichens

Expect

Geben Sie in Textform die Antwort ein, die der Server zurückgeben sollte, wenn er korrekt funktioniert. Falls Sie Ihr eigenes Sendeprogramm geschrieben haben, geben Sie die von Ihnen vorgegebene Antwort ein, die zu senden ist, falls es erfolgreich war.



Anmerkung

Um festzulegen, was für einen bestimmten Dienst gesendet werden sollte, können Sie eine **telnet**-Verbindung mit dem Port auf einem realen Server öffnen und sich anschauen, was zurückgegeben wird. Beispielsweise gibt FTP bei der Verbindung 220 aus, somit könnten Sie **quit** im Feld **Send** und **220** im Feld **Expect** eingeben.

Warnung

Vergessen Sie nicht, auf die Schaltfläche **ACCEPT** zu klicken, um sicherzustellen, dass keine Änderungen verloren gehen, wenn Sie den nächsten Reiter anklicken.

Nachdem Sie mithilfe des **Piranha-Konfigurationstools** virtuelle Server konfiguriert haben, müssen Sie bestimmte Konfigurationsdateien auf den Backup-LVS-Router kopieren. Siehe Abschnitt 4.7, „Synchronisieren von Konfigurationsdateien“ für Einzelheiten.

4.7. Synchronisieren von Konfigurationsdateien

Nach der Konfiguration des primären LVS-Routers gibt es mehrere Konfigurationsdateien, die auf den Backup-LVS-Router kopiert werden müssen, bevor Sie mit dem Load Balancer Add-On beginnen.

Bei diesen Dateien handelt es sich um:

- **/etc/sysconfig/ha/lvs.cf** – die Konfigurationsdatei für die LVS-Router.
- **/etc/sysctl** – die Konfigurationsdatei, die unter anderem die Paketweiterleitung im Kernel aktiviert.
- **/etc/sysconfig/iptables** – Falls Sie Firewall-Markierungen verwenden, sollten Sie eine dieser Dateien synchronisieren basierend darauf, welchen Netzwerkpaketfilter Sie verwenden.

Wichtig

Die Dateien **/etc/sysctl.conf** und **/etc/sysconfig/iptables** ändern sich *nicht*, wenn Sie das Load Balancer Add-On mithilfe des **Piranha-Konfigurationstools** konfigurieren.

4.7.1. Synchronisieren von lvs.cf

Jedes Mal, wenn die LVS-Konfigurationsdatei **/etc/sysconfig/ha/lvs.cf** erstellt oder verändert wird, muss sie auf den Backup-LVS-Routerknoten kopiert werden.

Warnung

Der aktive und der Backup-LVS-Routerknoten müssen über identische **lvs.cf**-Dateien verfügen. Abweichende LVS-Konfigurationsdateien zwischen den LVS-Routerknoten können zum Scheitern der Ausfallsicherung führen.

Am besten erreichen Sie dies mit dem **scp**-Befehl.

Wichtig

Um **scp** zu verwenden, muss **sshd** auf dem Backup-Router laufen. Siehe Abschnitt 2.1, „Konfigurieren von Diensten auf dem LVS-Router“ für Einzelheiten darüber, wie die notwendigen Dienste auf den LVS-Routern konfiguriert werden.

Führen Sie den folgenden Befehl als root-Benutzer auf dem primären LVS-Router aus, um die **lvs.cf**-Dateien zwischen den Routerknoten zu synchronisieren:

```
scp /etc/sysconfig/ha/lvs.cf n.n.n.n:/etc/sysconfig/ha/lvs.cf
```

Ersetzen Sie in dem obigen Befehl *n.n.n.n* durch die reale IP-Adresse des Backup-LVS-Routers.

4.7.2. Synchronisieren von sysctl

Die **sysctl**-Datei wird in den meisten Fällen nur einmal verändert. Diese Datei wird beim Systemstart gelesen und weist den Kernel dazu an, die Paketweiterleitung zu aktivieren.

Wichtig

Falls Sie sich nicht sicher sind, ob die Paketweiterleitung im Kernel aktiviert ist oder nicht, finden Sie unter Abschnitt 2.5, „Aktivieren der Paketweiterleitung“ Anweisungen, wie Sie diese wichtige Einstellung überprüfen und gegebenenfalls anpassen können.

4.7.3. Synchronisieren von Regeln zur Netzwerkpaketfilterung

Falls Sie **iptables** verwenden, müssen Sie die entsprechende Konfigurationsdatei auf den Backup-LVS-Router synchronisieren.

Falls Sie jegliche der Netzwerkpaketfilter-Regeln ändern, führen Sie den folgenden Befehl als root-Benutzer auf dem primären LVS-Router aus:

```
scp /etc/sysconfig/iptables n.n.n.n:/etc/sysconfig/
```

Ersetzen Sie in dem obigen Befehl *n.n.n.n* durch die reale IP-Adresse des Backup-LVS-Routers.

Öffnen Sie als Nächstes eine **ssh**-Sitzung zum Backup-Router oder melden Sie sich auf dem Rechner als root-Benutzer an und führen Sie den folgenden Befehl aus:

```
/sbin/service iptables restart
```

Sobald Sie diese Dateien auf den Backup-Router übertragen und die erforderlichen Dienste gestartet haben (siehe Abschnitt 2.1, „Konfigurieren von Diensten auf dem LVS-Router“), sind Sie nun dazu

bereit, das Load Balancer Add-On zu starten.

4.8. Starten des Load Balancer Add-Ons

Zum Starten des Load Balancer Add-Ons ist es am besten, gleichzeitig zwei root-Terminals zu öffnen oder gleichzeitig zwei **ssh**-Sitzungen als root zum primären LVS-Router herzustellen.

Beobachten Sie in dem einen Terminal die Kernel-Protokollmeldungen mithilfe des folgenden Befehls:

```
tail -f /var/log/messages
```

Starten Sie anschließend das Load Balancer Add-On, indem Sie den folgenden Befehl in dem zweiten Terminal ausführen:

```
/sbin/service pulse start
```

Folgen Sie dem Fortschritt des **pulse**-Dienststarts in dem Terminal mit den Kernel-Protokollnachrichten. Wenn Sie die folgende Ausgabe sehen, wurde der **pulse**-Dienst erfolgreich gestartet:

```
gratuitous lvs arps finished
```

Um die Nachverfolgung von **/var/log/messages** zu beenden, drücken Sie **Strg**+**c**.

Von diesem Zeitpunkt an ist der primäre LVS-Router auch der aktive LVS-Router. Sie können zwar bereits Anfragen an das Load Balancer Add-On stellen, sollten jedoch vor Inbetriebnahme auch den Backup-LVS-Router starten. Wiederholen Sie dazu einfach das eben beschriebene Vorgehen auf dem Backup-LVS-Routerknoten.

Sobald Sie diesen letzten Schritt durchgeführt haben, ist das Load Balancer Add-On aktiv und einsatzbereit.

Verwenden des Load Balancer Add-Ons mit dem High Availability Add-On

Sie können das Load Balancer Add-On mit dem High Availability Add-On verwenden, um eine hochverfügbare E-Commerce-Lösung bereitzustellen, die Lastverteilung, Datenintegrität und eine hohe Applikationsverfügbarkeit bietet.

Die Konfiguration in Abbildung A.1, „Load Balancer Add-On mit einem High Availability Add-On“ zeigt eine E-Commerce-Lösung zur Warenbestellung über eine URL. Client-Anfragen an die URL durchqueren die Firewall zum aktiven LVS-Lastverteilungsrouter, der diese Anfragen dann an einen der Webserver weiterleitet. Die High Availability Add-On Knoten liefern dynamische Daten an die Webserver, die diese Daten wiederum an den anfragenden Client weiterleiten.

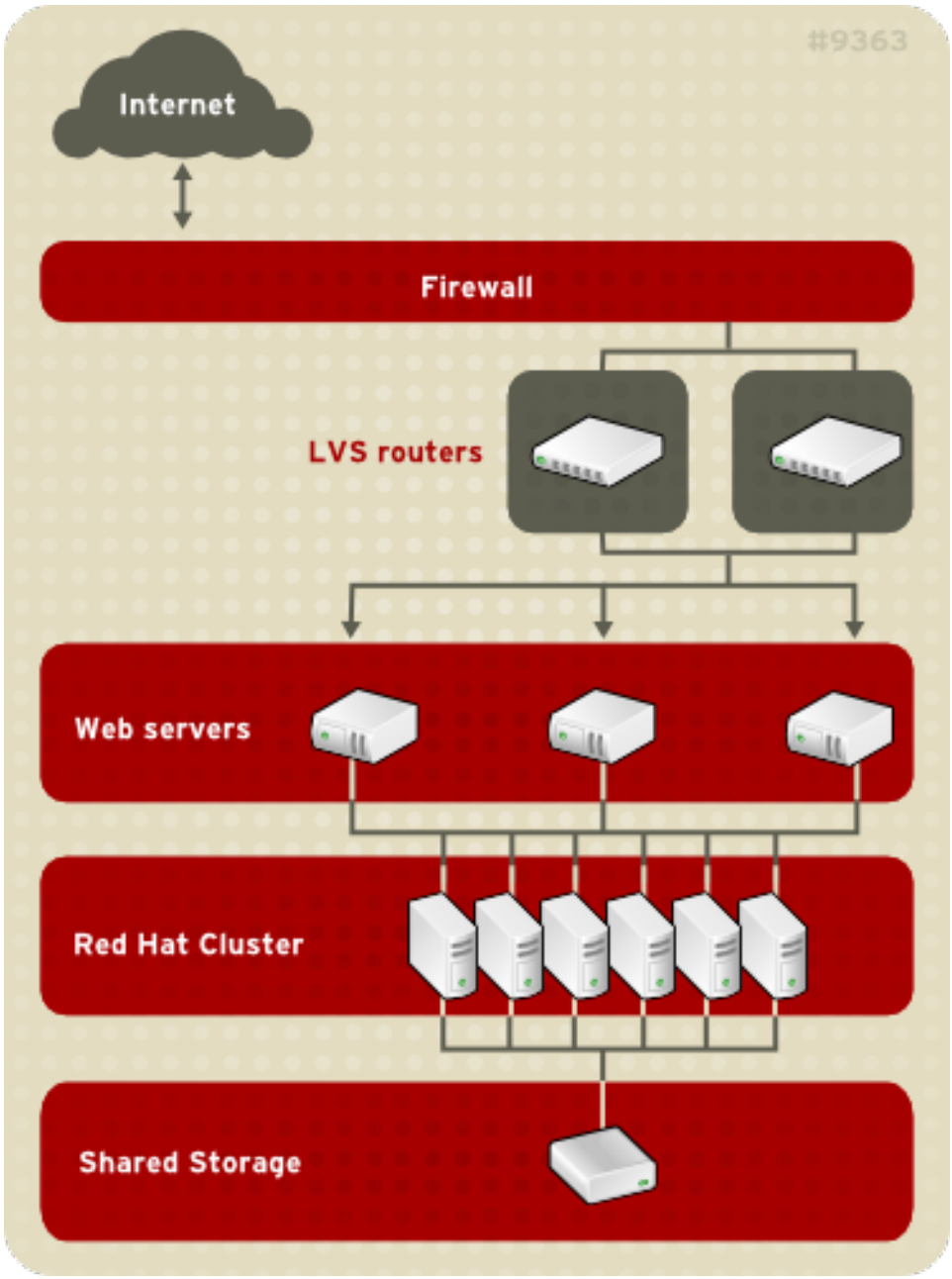


Abbildung A.1. Load Balancer Add-On mit einem High Availability Add-On

Das Bereitstellen dynamischer Webinhalte mit dem Load Balancer Add-On erfordert eine dreischichtige Konfiguration (siehe Abbildung A.1, „Load Balancer Add-On mit einem High Availability Add-On“). Diese Kombination aus Load Balancer Add-On und High Availability Add-On ermöglicht die Konfiguration einer hochintegrierten E-Commerce-Lösung ohne Single Point of Failure. Das High Availability Add-On kann eine hochverfügbare Instanz einer Datenbank oder einer Reihe von Datenbanken ausführen, die über das Netzwerk für die Webserver erreichbar sind.

Eine dreischichtige Konfiguration ist erforderlich, um dynamische Inhalte bereitzustellen. Eine zweischichtige Load Balancer Add-On Konfiguration ist geeignet für Webserver, die lediglich statische Webinhalte bereitstellen (mit nur geringen und seltenen Änderungen). Eine zweischichtige Konfiguration ist jedoch nicht geeignet, falls die Webserver dynamische Inhalte bereitstellen. Zu dynamischen Inhalten gehören Produktinventare, Bestellanforderungen oder Kundendatenbanken, die auf allen Webservern konsistent sein müssen, um zu gewährleisten, dass Kunden Zugriff auf aktuelle und korrekte Informationen haben.

Die einzelnen Schichten haben die folgenden Funktionen:

- Erste Schicht – LVS-Router führen die Lastverteilung aus, um Webanfragen zu verteilen.
- Zweite Schicht – Eine Reihe von Webservern antwortet auf diese Anfragen.
- Dritte Schicht – Ein High Availability Add-On liefert Daten an die Webserver.

In einer Load Balancer Add-On Konfiguration wie der in Abbildung A.1, „Load Balancer Add-On mit einem High Availability Add-On“ tätigen Client-Systeme Anfragen im World Wide Web. Aus Sicherheitsgründen gehen diese Anfragen durch eine Firewall bei einer Website ein. Bei der Firewall kann es sich um ein Linux-System handeln, das diesem Zweck dient, oder um ein dediziertes Firewall-Gerät. Zwecks Redundanz können Sie Firewall-Geräte in einer Konfiguration zur Ausfallsicherung konfigurieren. Hinter der Firewall befindet sich ein LVS-Router, der die Lastverteilung durchführt, die in einem Aktiv-Standby-Modus konfiguriert werden kann. Der aktive Lastverteilungsrouter leitet die Anfragen an die Webserver weiter.

Jeder Webserver kann unabhängig eine HTTP-Anfrage von einem Client empfangen und die Antwort an den Client senden. Das Load Balancer Add-On ermöglicht es Ihnen, die Kapazität einer Website zu erhöhen, indem Sie Webserver hinter dem LVS-Router hinzufügen; der LVS-Router führt die Lastverteilung dann auf einer größeren Gruppe von Webservern aus. Falls ein Webserver ausfällt, kann dieser entfernt werden; das Load Balancer Add-On führt daraufhin die Lastverteilung auf einer kleineren Gruppe von Webservern fort.

Versionsgeschichte

Version 1-15.3 **Thu Apr 9 2015** **Hedda Peters**
Deutsche Übersetzung des Handbuchtitels geändert

Version 1-15.2 **Thu Apr 9 2015** **Hedda Peters**
Deutsche Übersetzung fertiggestellt

Version 1-15.1 **Thu Apr 9 2015** **Hedda Peters**
Übersetzungsdateien synchronisiert mit XML-Quellen 1-15

Version 1-15 **Tue Dec 16 2014** **Steven Levine**
Implementierung von sort_order auf der RHEL 6 Splash-Seite.

Version 1-13 **Thu Oct 9 2014** **Steven Levine**
Release für GA von Red Hat Enterprise Linux 6.6

Version 1-10 **Tue Nov 19 2013** **John Ha**
Release für Red Hat Enterprise Linux 6.5 Beta

Version 1-8 **Fri Sep 27 2013** **John Ha**
Release für Red Hat Enterprise Linux 6.5 Beta

Version 1-4 **Wed Nov 28 2012** **John Ha**
Release für Red Hat Enterprise Linux 6.4 Beta

Version 1-3 **Mon Jun 18 2012** **John Ha**
Release für GA von Red Hat Enterprise Linux 6.3

Version 1-2 **Fri Dec 2 2011** **John Ha**
Release für GA von Red Hat Enterprise Linux 6.2

Version 1-1 **Wed Nov 10 2010** **Paul Kennedy**
Erste Release des Handbuchs für Red Hat Enterprise Linux 6

Stichwortverzeichnis

Symbole

/etc/sysconfig/ha/lvs.cf-Datei, /etc/sysconfig/ha/lvs.cf

A

arptables_jf, Direktes Routing und arptables_jf

C

chkconfig, Konfigurieren von Diensten auf dem LVS-Router

Cluster
- Verwenden des Load Balancer Add-Ons mit dem High Availability Add-On, Verwenden des Load Balancer Add-Ons mit dem High Availability Add-On

D

direktes Routing
- und arptables_jf, Direktes Routing und arptables_jf

E

Einführung, Einführung
- weitere Red Hat Enterprise Linux Dokumente, Einführung

F

Feedback, Feedback

FTP, Konfigurieren von FTP
- (Siehe auch Load Balancer Add-On)

H

High Availability Add-On
- und das Load Balancer Add-On , Verwenden des Load Balancer Add-Ons mit dem High Availability Add-On
- Verwenden mit dem Load Balancer Add-On, Verwenden des Load Balancer Add-Ons mit dem High Availability Add-On

I

iptables , Konfigurieren von Diensten auf dem LVS-Router

ipvsadm-Programm, ipvsadm

J

Job-Scheduling, Load Balancer Add-On, Übersicht über das Load Balancer Add-On Scheduling

K

Komponenten
- Load Balancer Add-On, Load Balancer Add-On Komponenten

L

Least Connections (Siehe Job-Scheduling, Load Balancer Add-On)

Least Connections, gewichtet (Siehe Job-Scheduling, Load Balancer Add-On)

Load Balancer Add-On
- /etc/sysconfig/ha/lvs.cf-Datei, /etc/sysconfig/ha/lvs.cf
- Datenreplikation, reale Server, Daten replizieren und gemeinsam verwenden auf realen Servern
- direktes Routing
  - Anforderungen, Hardware, Direktes Routing, Load Balancer Add-On mit direktem Routing
  - Anforderungen, Netzwerk, Direktes Routing, Load Balancer Add-On mit direktem Routing
  - Anforderungen, Software, Direktes Routing, Load Balancer Add-On mit direktem Routing
  - und arptables_jf, Direktes Routing und arptables_jf
- dreischichtig
  - Load Balancer Add-on, Eine dreischichtige Load Balancer Add-On Konfiguration

- erste Konfigurationsschritte, Erste Konfigurationsschritte für das Load Balancer Add-On
- gemeinsam verwendete Daten, Daten replizieren und gemeinsam verwenden auf realen Servern
- ipvsadm-Programm, ipvsadm
- Job-Scheduling, Übersicht über das Load Balancer Add-On Scheduling
- Komponenten, Load Balancer Add-On Komponenten
- LVS-Router
    - Konfigurieren von Diensten, Erste Konfigurationsschritte für das Load Balancer Add-On
    - Notwendige Dienste, Konfigurieren von Diensten auf dem LVS-Router
    - primärer Knoten, Erste Konfigurationsschritte für das Load Balancer Add-On
- Multi-Port-Dienste, Multi-Port-Dienste und Load Balancer Add-On
    - FTP, Konfigurieren von FTP
- nanny-Daemon, nanny
- NAT-Routing
    - Anforderungen, Hardware, Das NAT Load Balancer Add-On Netzwerk
    - Anforderungen, Netzwerk, Das NAT Load Balancer Add-On Netzwerk
    - Anforderungen, Software, Das NAT Load Balancer Add-On Netzwerk
- Paketweiterleitung, Aktivieren der Paketweiterleitung
- Piranha-Konfigurationstool , Piranha-Konfigurationstool
- pulse-Daemon, pulse
- Routing-Methoden
    - NAT, Routing-Methoden
- Routing-Voraussetzungen, Konfigurieren von Netzwerkschnittstellen für Load Balancer Add-On mit NAT
- Scheduling, Job, Übersicht über das Load Balancer Add-On Scheduling
- send_arp-Programm, send_arp
- Starten des Load Balancer Add-Ons, Starten des Load Balancer Add-Ons
- Synchronisieren von Konfigurationsdateien, Synchronisieren von Konfigurationsdateien
- Verwenden des Load Balancer Add-Ons mit dem High Availability Add-On, Verwenden des Load Balancer Add-Ons mit dem High Availability Add-On

LVS
- Daemon, lvs
- lvs-Daemon, lvs
- NAT-Routing
    - aktivieren, Aktivieren von NAT-Routing auf den LVS-Routern
- reale Server, Überblick über das Load Balancer Add-On
- Überblick, Überblick über das Load Balancer Add-On

lvs-Daemon, lvs

M

Multi-Port-Dienste, Multi-Port-Dienste und Load Balancer Add-On
- (Siehe auch Load Balancer Add-On)

N

nanny-Daemon, nanny

NAT
- aktivieren, Aktivieren von NAT-Routing auf den LVS-Routern
- Routing-Methoden, Load Balancer Add-On, Routing-Methoden

Network Address Translation (Siehe NAT)

P

Paketweiterleitung, Aktivieren der Paketweiterleitung
- (Siehe auch Load Balancer Add-On)

piranha-gui-Dienst, Konfigurieren von Diensten auf dem LVS-Router

Piranha-Konfigurationstool , Piranha-Konfigurationstool
- Anmeldeseite, Anmelden beim Piranha-Konfigurationstool
- CONTROL/MONITORING , CONTROL/MONITORING
- EDIT MONITORING SCRIPTS Unterabschnitt, Der Unterabschnitt EDIT MONITORING SCRIPTS
- Einrichten eines Passworts, Einrichten eines Passworts für das Piranha-Konfigurationstool
- GLOBAL SETTINGS , GLOBAL SETTINGS
- notwendige Software, Notwendige Software
- REAL SERVER Unterabschnitt , Der Unterabschnitt REAL SERVER
- REDUNDANCY , REDUNDANCY
- Überblick, Konfigurieren des Load Balancer Add-Ons mit dem Piranha-Konfigurationstool
- VIRTUAL SERVER Unterabschnitt, Der Unterabschnitt VIRTUAL SERVER
  - Firewall-Markierung , Der Unterabschnitt VIRTUAL SERVER
  - Persistence , Der Unterabschnitt VIRTUAL SERVER
  - Scheduling , Der Unterabschnitt VIRTUAL SERVER
  - Virtuelle IP-Adresse , Der Unterabschnitt VIRTUAL SERVER
- VIRTUAL SERVERS , VIRTUAL SERVERS
- Zugriff einschränken, Einschränken des Zugriffs auf das Piranha-Konfigurationstool

piranha-passwd , Einrichten eines Passworts für das Piranha-Konfigurationstool

pulse-Daemon, pulse

pulse-Dienst, Konfigurieren von Diensten auf dem LVS-Router

R

reale Server
- Konfigurieren von Diensten, Konfigurieren von Diensten auf den realen Servern

Round Robin (Siehe Job-Scheduling, Load Balancer Add-On)

Round Robin, gewichtet (Siehe Job-Scheduling, Load Balancer Add-On)

Routing
- Voraussetzungen für Load Balancer Add-On, Konfigurieren von Netzwerkschnittstellen für Load Balancer Add-On mit NAT

S

Scheduling, Job (Load Balancer Add-On), Übersicht über das Load Balancer Add-On Scheduling

send_arp-Programm, send_arp

Sicherheit

- Piranha-Konfigurationstool, Einschränken des Zugriffs auf das Piranha-Konfigurationstool

sshd-Dienst, Konfigurieren von Diensten auf dem LVS-Router

Synchronisieren von Konfigurationsdateien, Synchronisieren von Konfigurationsdateien